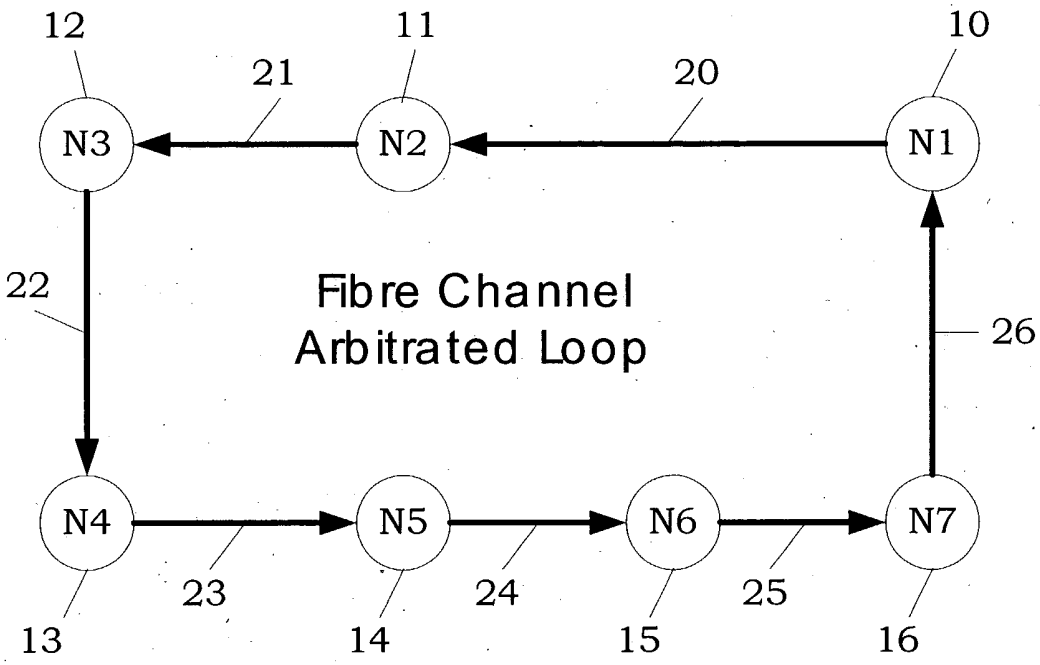

12
11
10
21
20
N3
N2
N1
22
Fibre Channel
Arbitrated Loop
26
N4
N5
N6
N7
23
24
25
13
14
15
16

Fig. 1

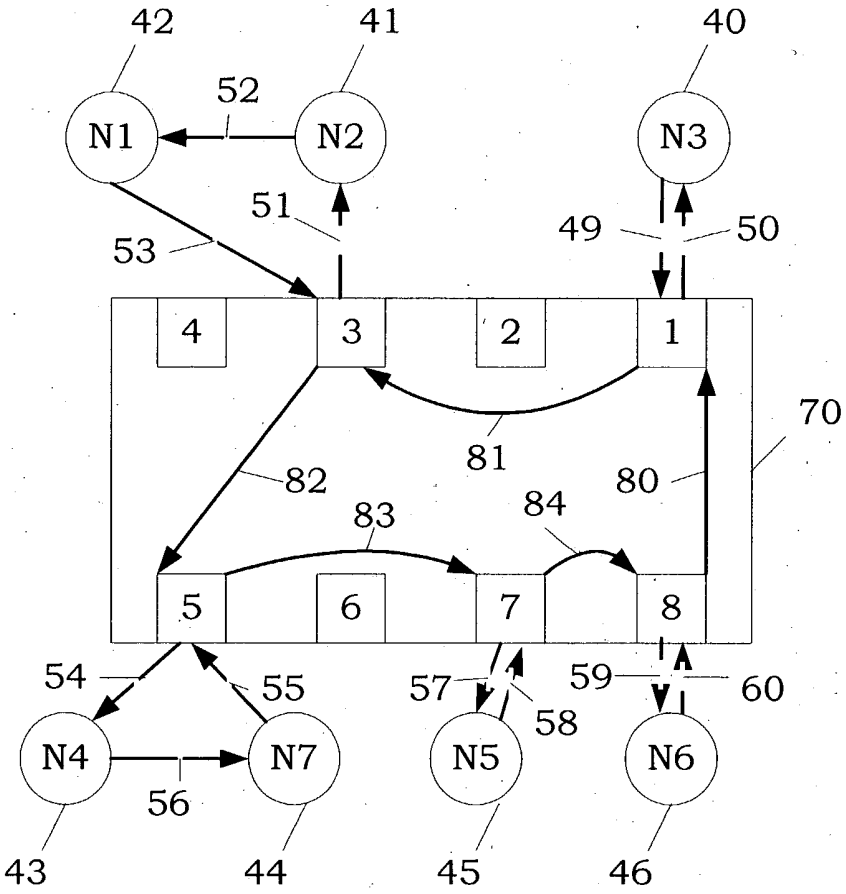
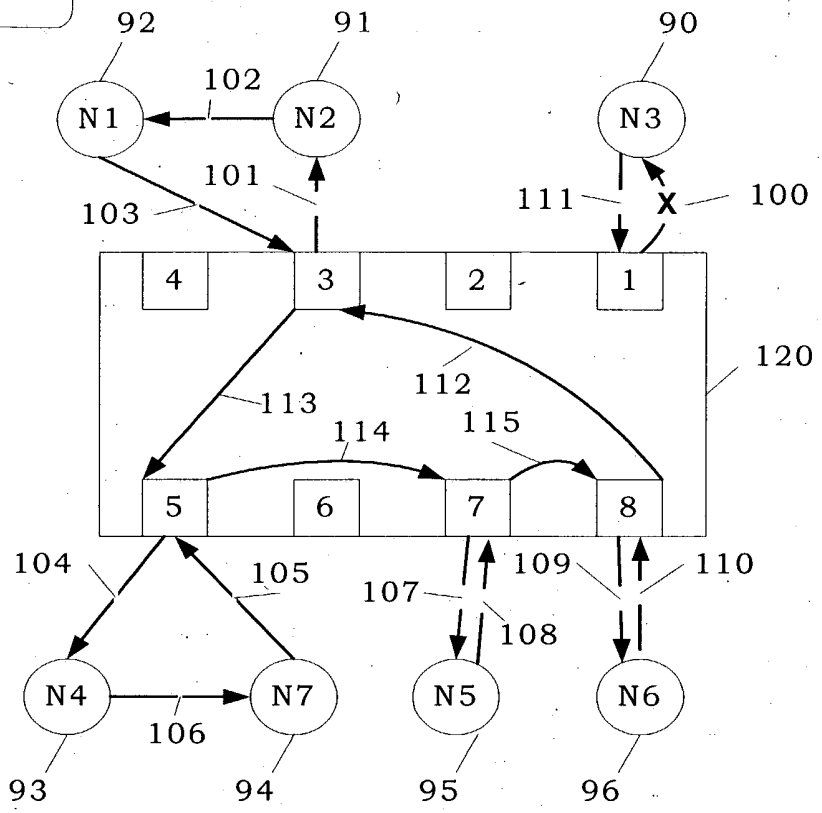

42
41
40
52
N1
N2
N3
51
49
50
53
4
3
2
1
70
81
82
80
83
84
5
6
7
8
54
55
57
59
60
58
N4
N7
N5
N6
56
43
44
45
46

Fig. 2



Fibre Channel Arbitrated Loop Hub

Fig. 3

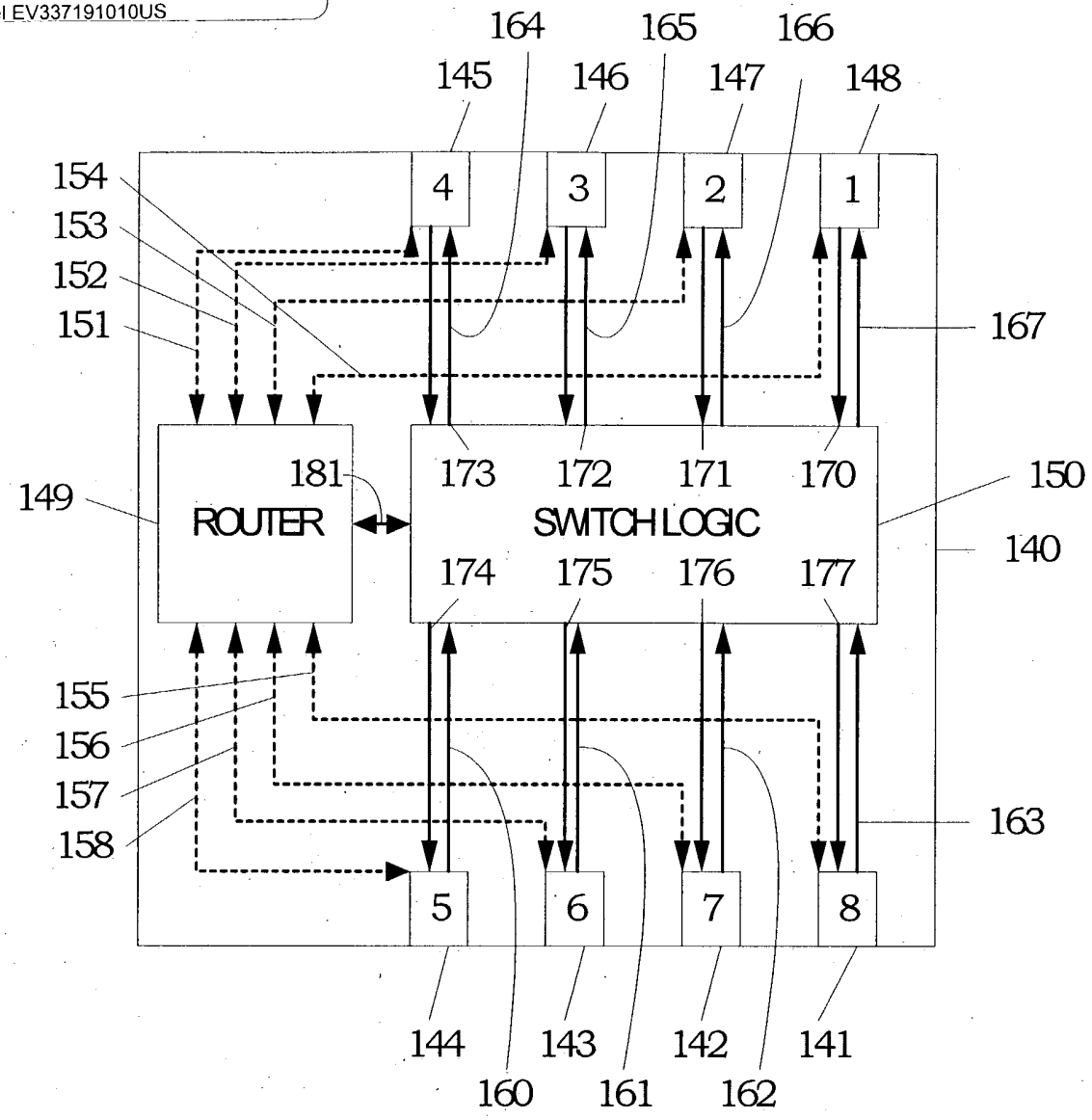

164
165
166
145
146
147
148
154
153
152
151
4
3
2
1
167
149
181
173
172
171
170
150
ROUTER
SWITCH LOGIC
140
174
175
176
177
155
156
157
158
163
5
6
7
8
144
143
142
141
160
161
162

Fig. 4

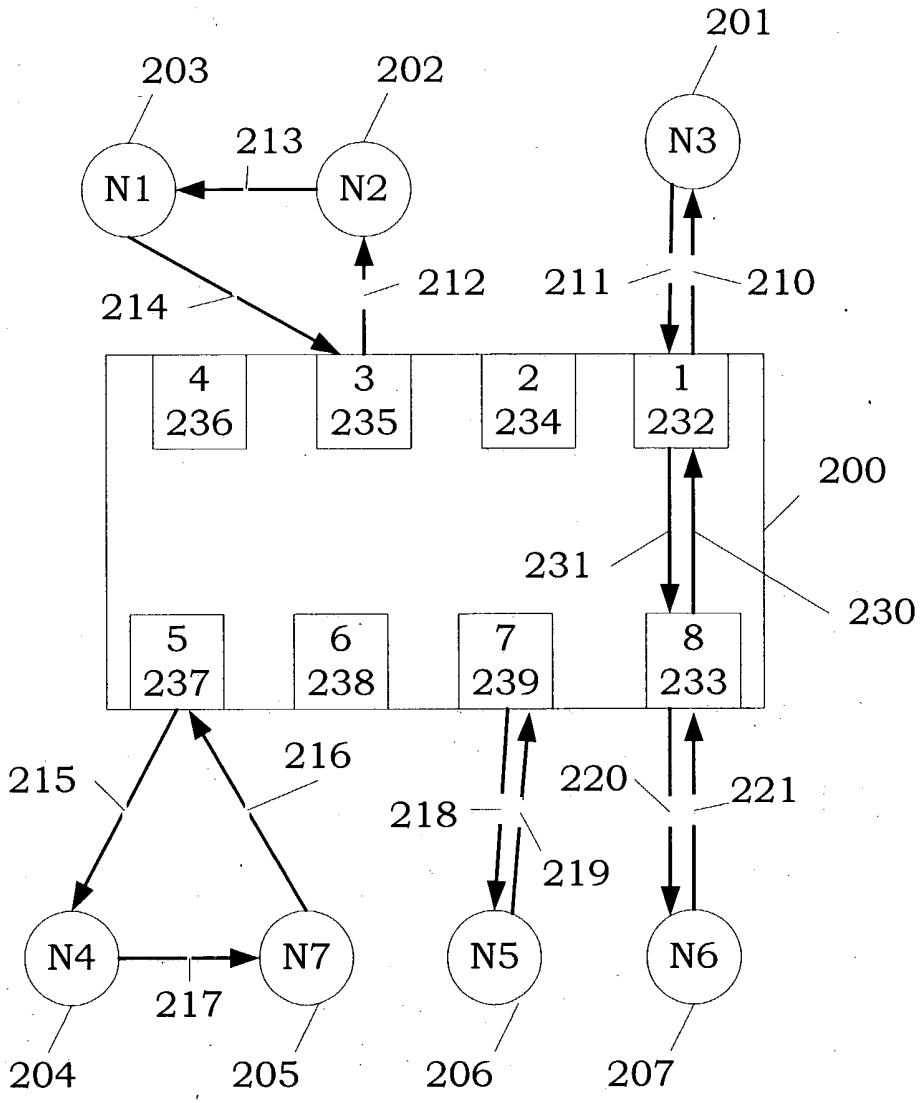

201
203
202
213
N1
N2
N3
214
212
211
210
4
236
3
235
2
234
1
232
200
231
230
5
237
6
238
7
239
8
233
215
216
218
220
221
219
N4
N7
N5
N6
217
204
205
206
207

Fig. 5

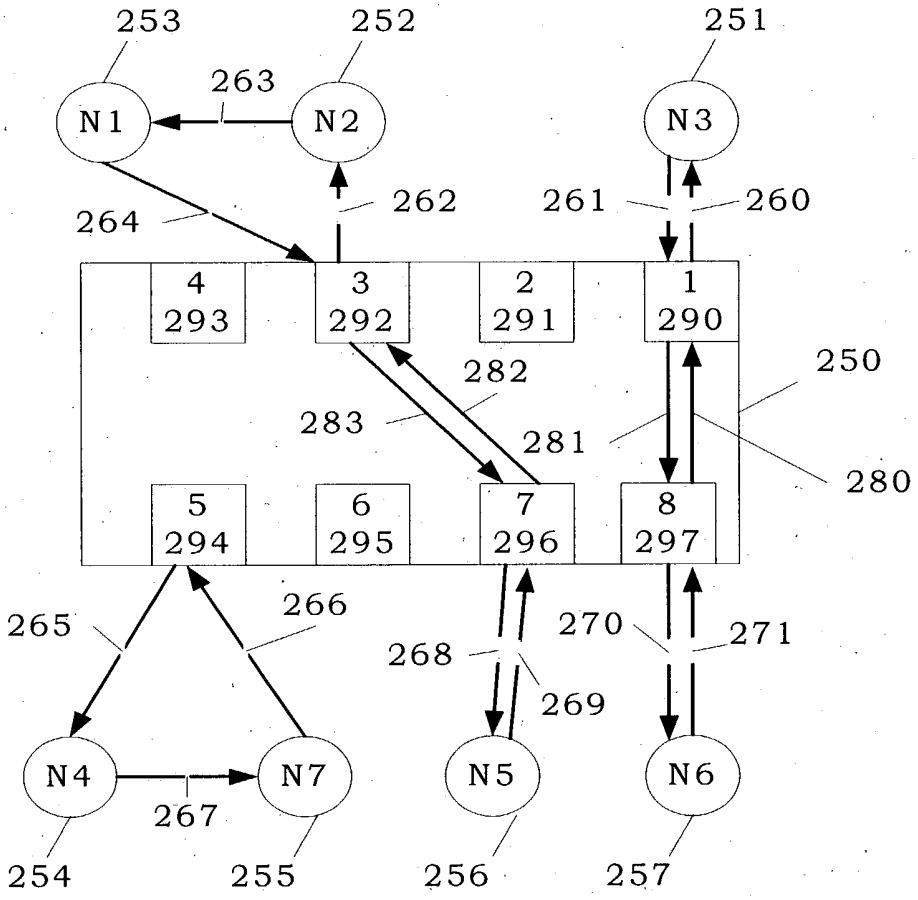

253
252
251
263
N1
N2
N3
264
262
261
260
4
293
3
292
2
291
1
290
282
250
283
281
280
5
294
6
295
7
296
8
297
265
266
268
270
271
269
N4
N7
N5
N6
267
254
255
256
257

Fig. 6

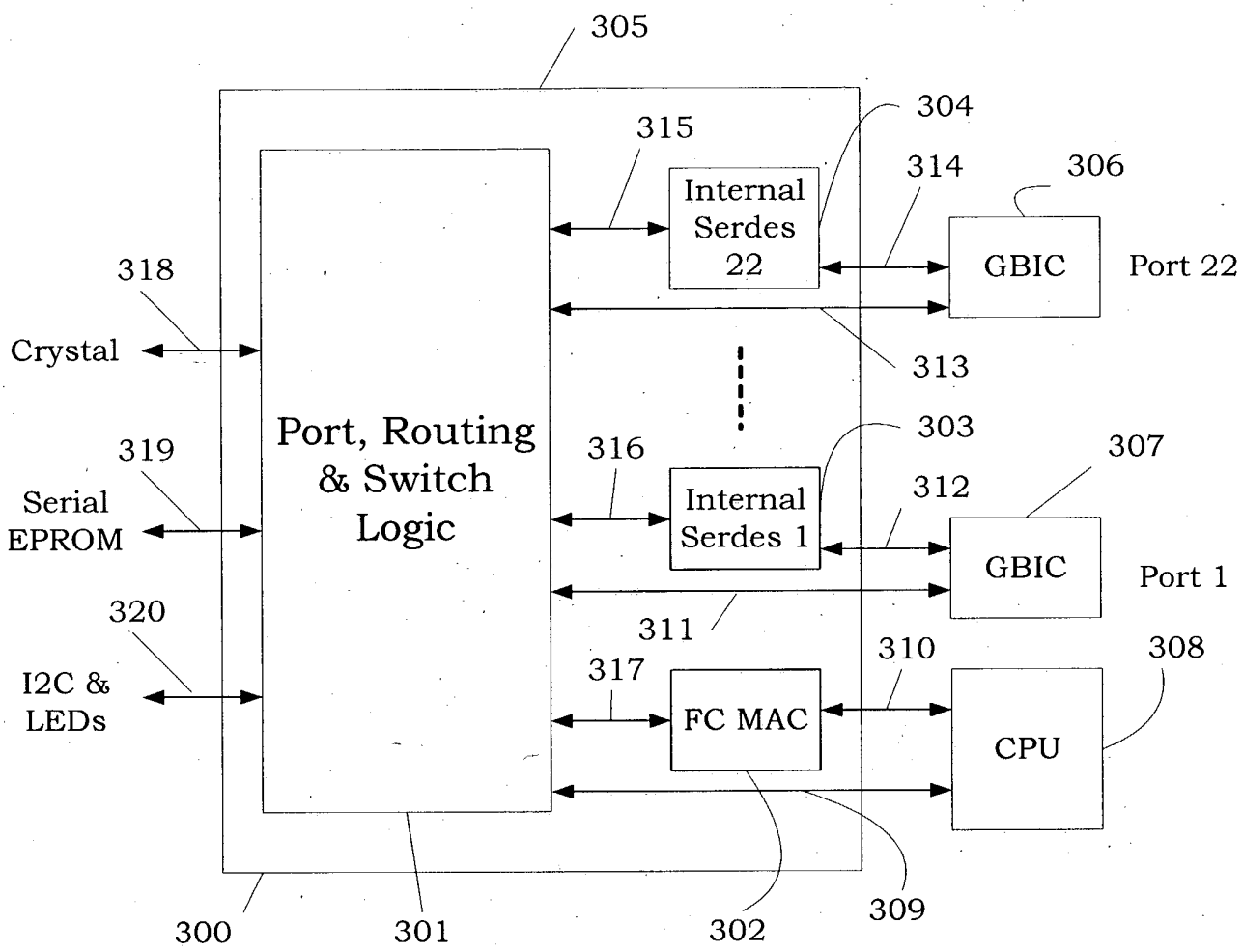

305
315
304
314
306
Internal Serdes 22
GBIC
Port 22
318
Crystal
313
Port, Routing & Switch Logic
303
316
307
319
Serial EPROM
Internal Serdes 1
312
GBIC
Port 1
311
320
310
308
317
I2C & LEDs
FC MAC
CPU
309
300
301
302

Fig. 7

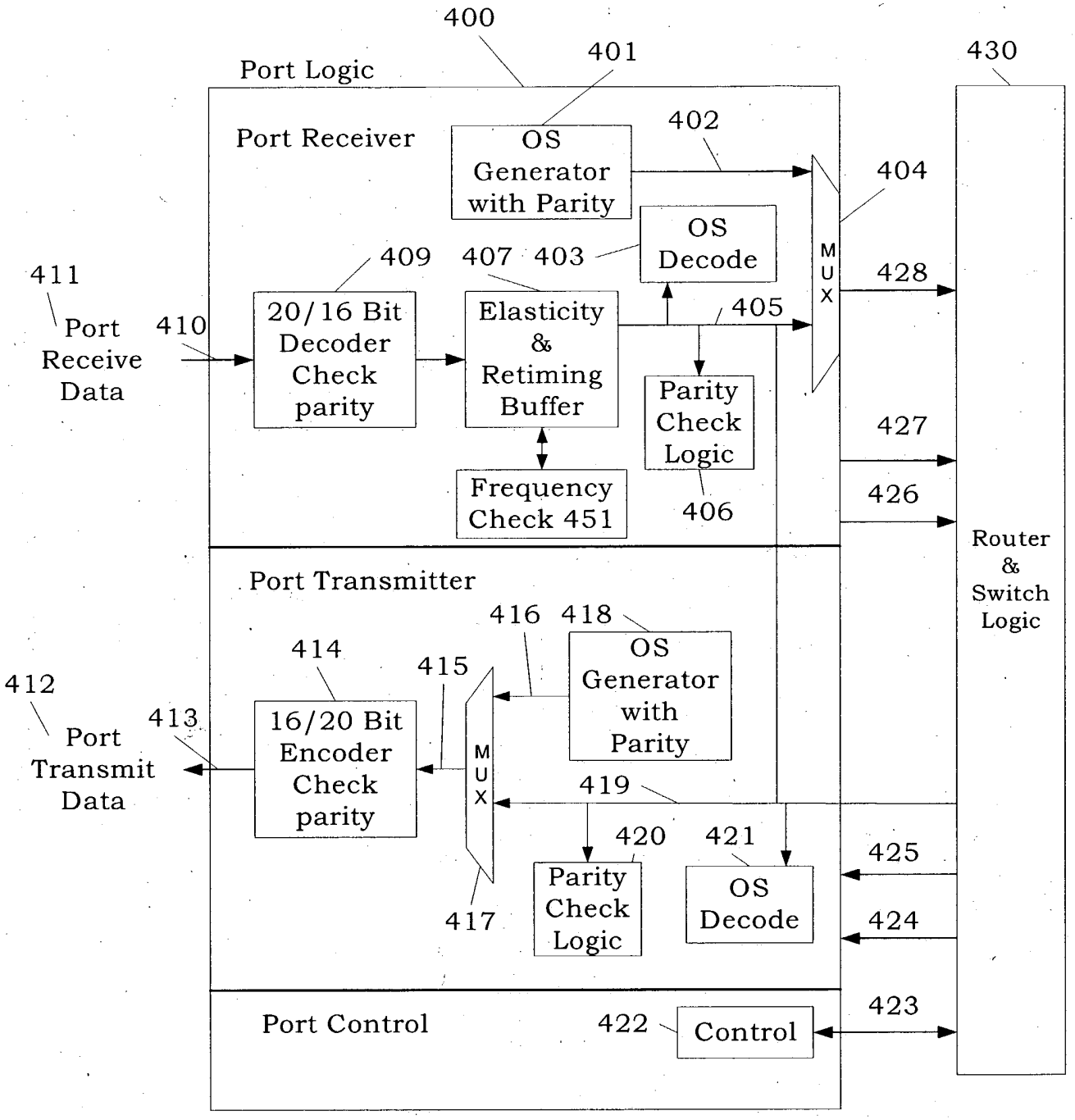

400
401
430
Port Logic
Port Receiver
OS Generator with Parity
402
404
OS Decode
409
407
403
411
Port Receive Data
410
20/16 Bit Decoder Check parity
Elasticity & Retiming Buffer
405
MUX
428
Parity Check Logic
427
Frequency Check 451
406
426
Router & Switch Logic
Port Transmitter
416
418
OS Generator with Parity
414
415
412
Port Transmit Data
413
16/20 Bit Encoder Check parity
MUX
419
420
421
425
Parity Check Logic
OS Decode
417
424
Port Control
422
Control
423

Fig. 8

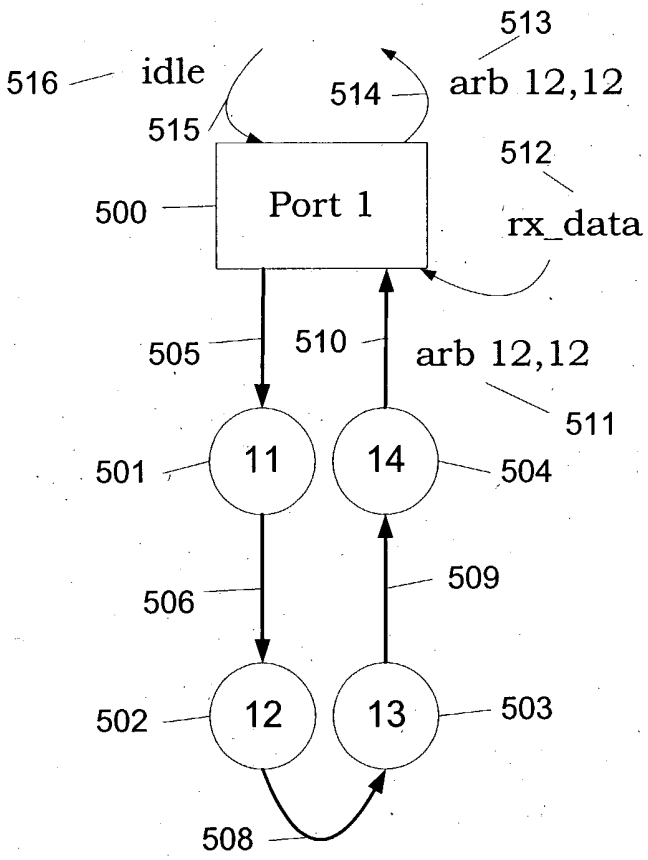
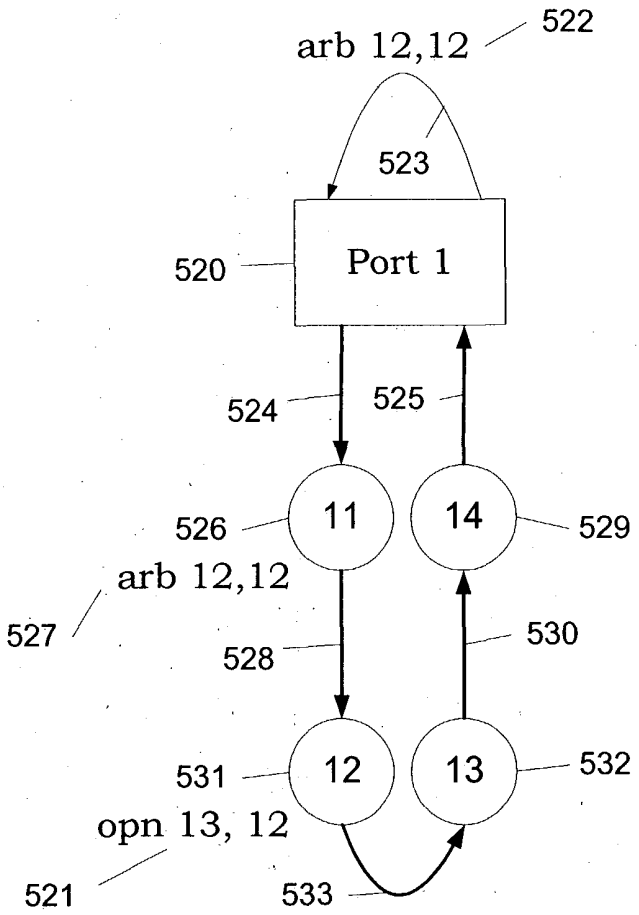
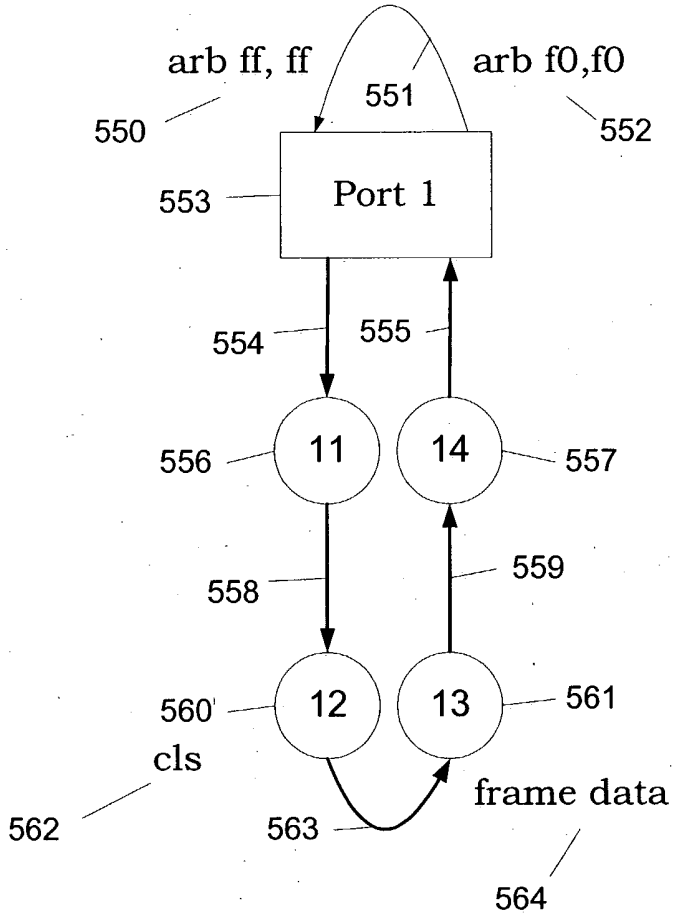

513
516
idle
514
arb 12,12
515
512
500
Port 1
rx_data
505
510
arb 12,12
511
501
11
14
504
506
509
502
12
13
503
508
Arb detected on
port rx_data

Fig 9a



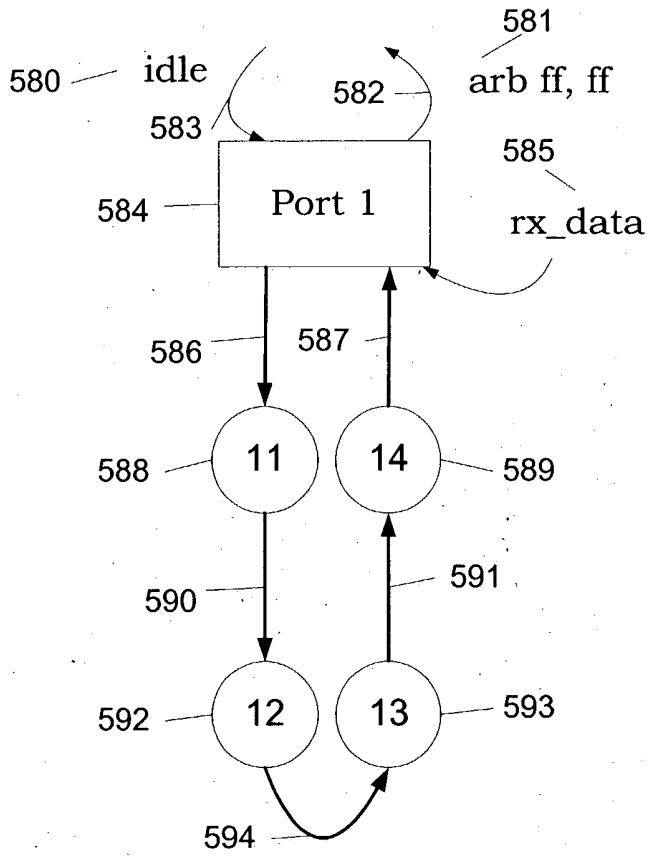
Connection established.
Open sourced

Fig 9b



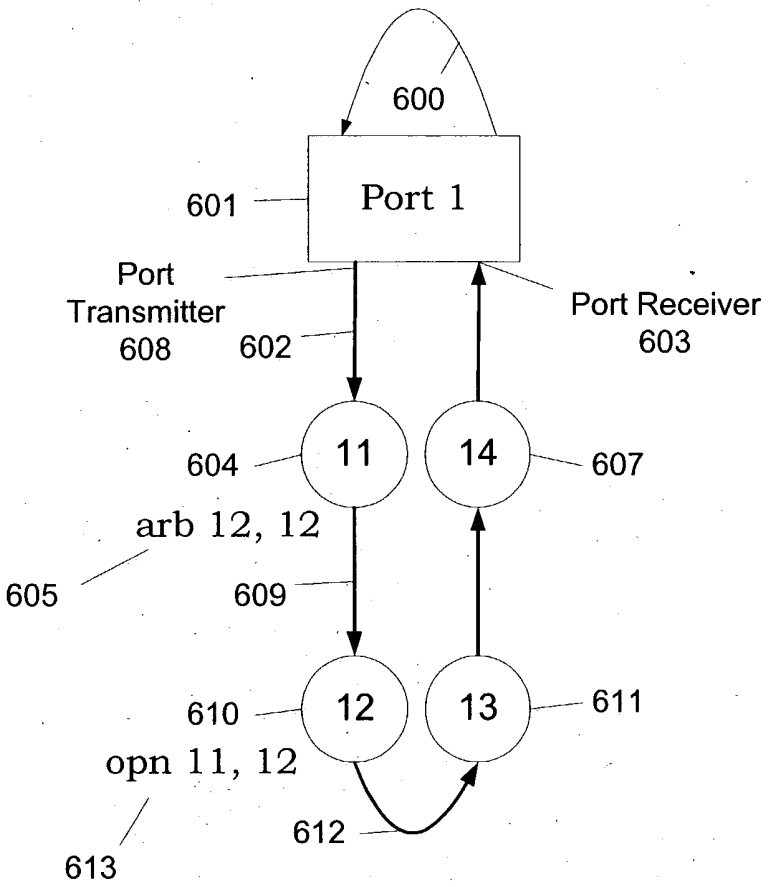
Arbf0 replaced with Arbff. Data Transfered.
Close sourced.

Fig 9c



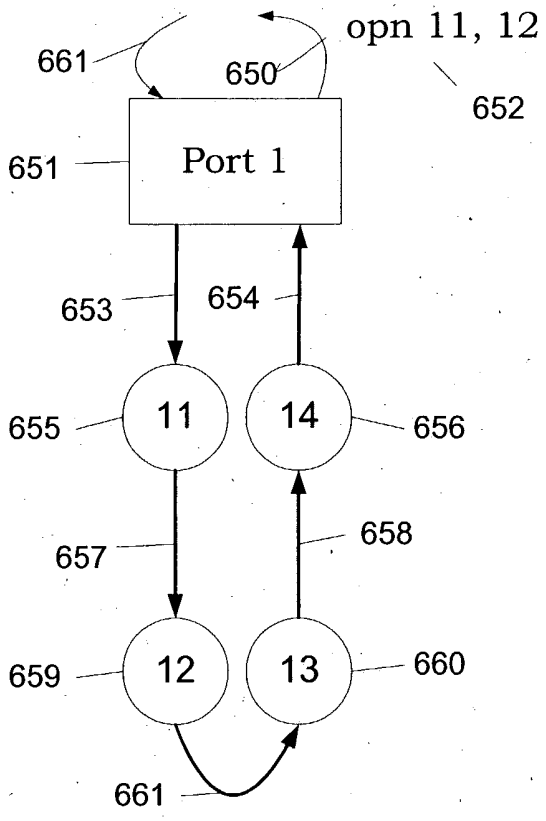
Arbff detected on port rx_data.
Disconnect. Source idles

Fig 9d



Arbs forwarded. Open sourced.

Fig 10a



Open detected at port receiver.
Port receiver holds open.
Provides ALPA to router and asserts opn_connect_req.

Fig 10b

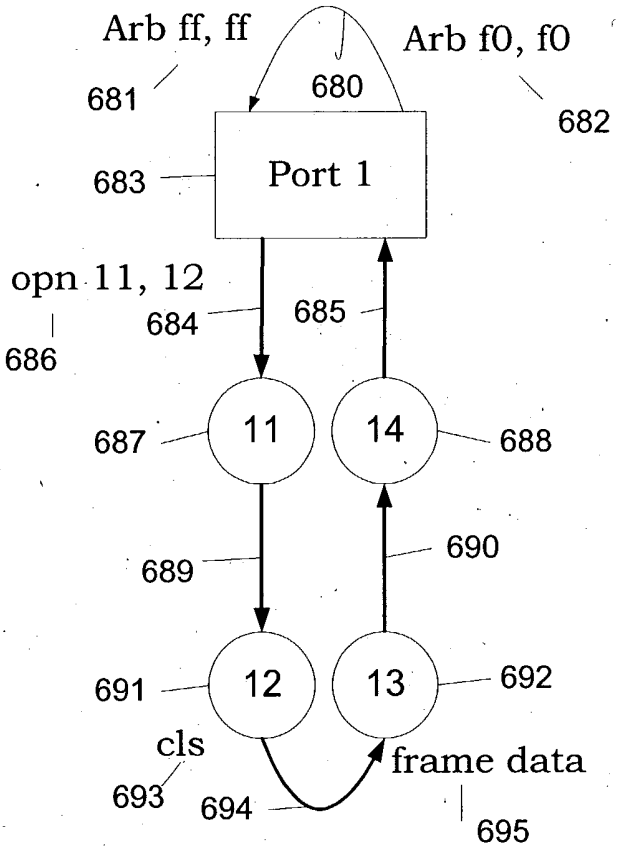
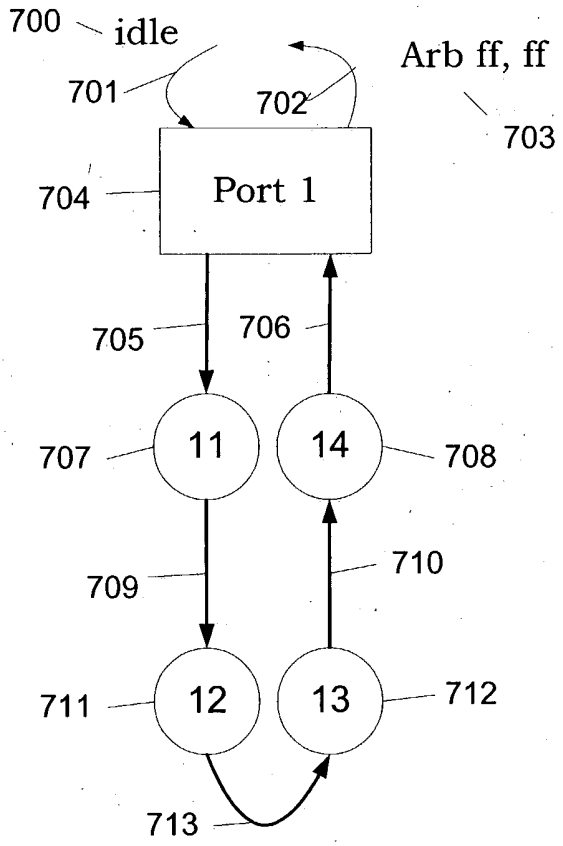
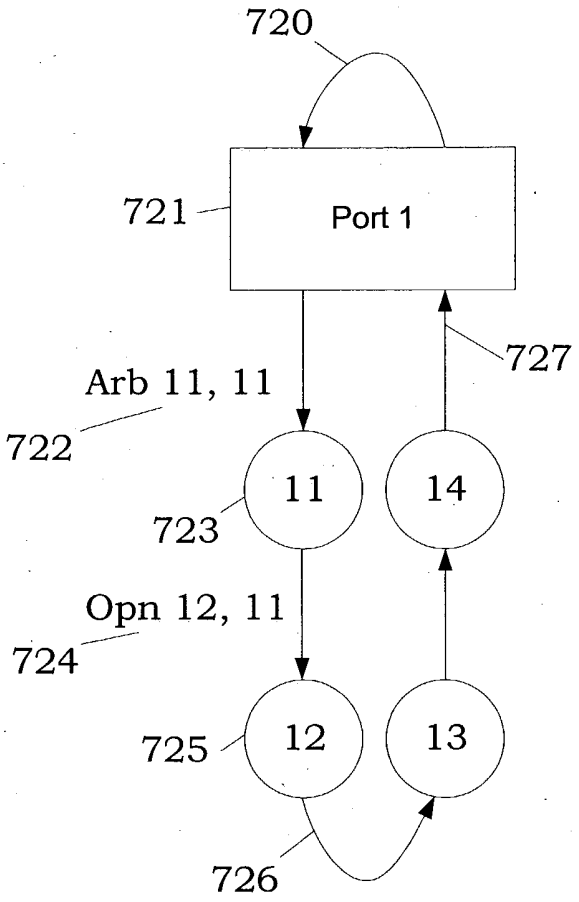

Arb ff, ff
681
680
Arb f0, f0
682
Port 1
683
opn 11, 12
684
685
686
687
11
14
688
689
690
691
12
13
692
cls
693
694
frame data
695

Fig 10c



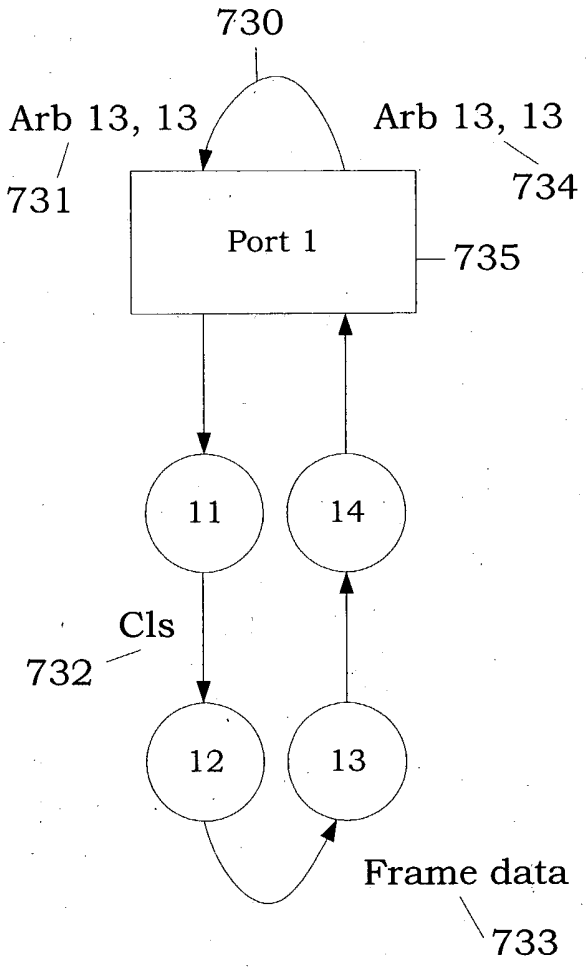
Arbff detected on port rx_data. Disconnect. Source idles.

Fig 10d



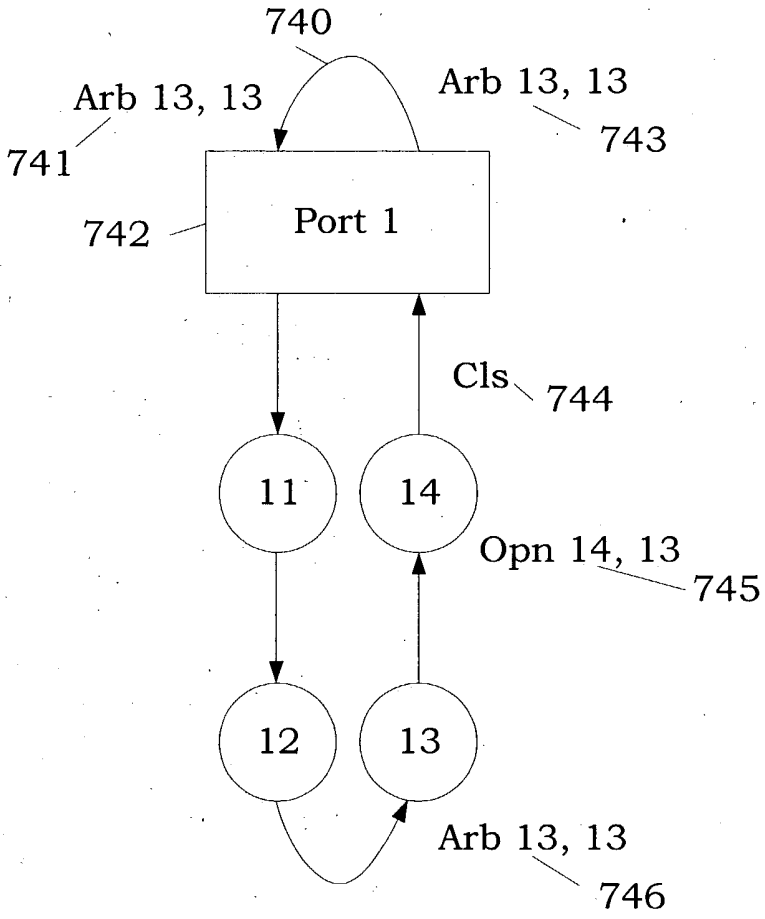
Arbs forwarded.
Open Sourced.

Fig 11a



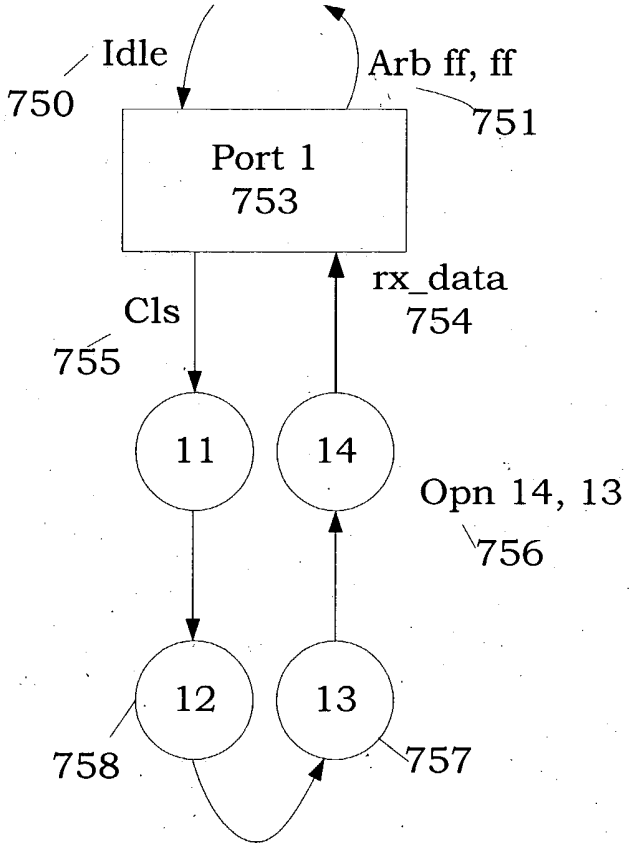
Arb 13 forwarded.
Data Transfered.
Close sourced.

Fig 11b



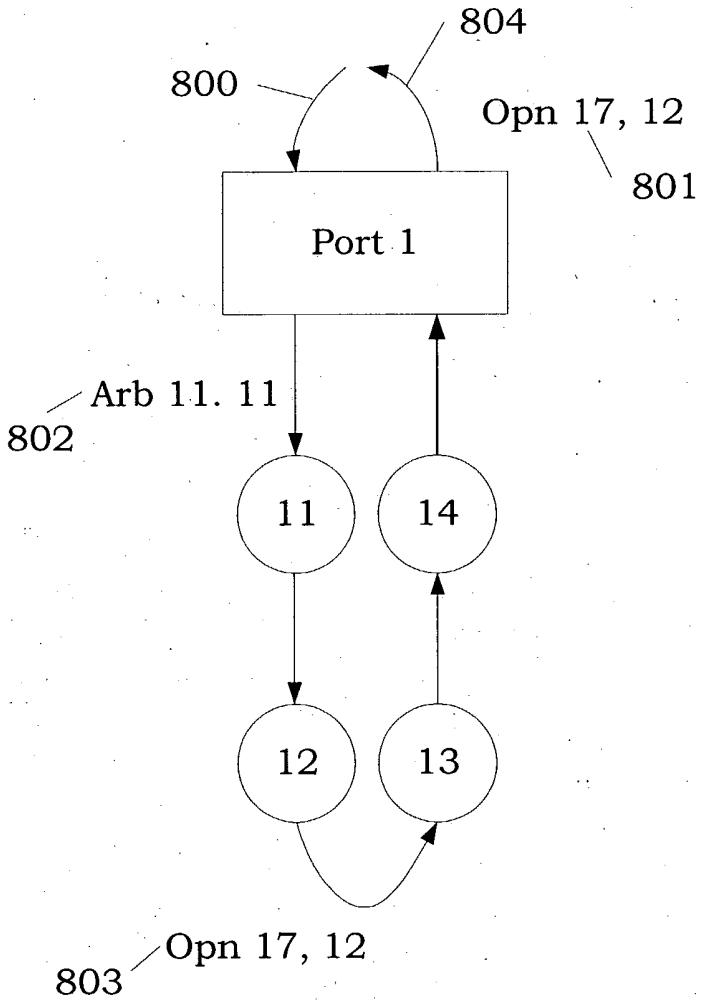
Close forwarded by node 12.
Arb 13 received by node 13.
Open sourced by node 13.

Fig 11c



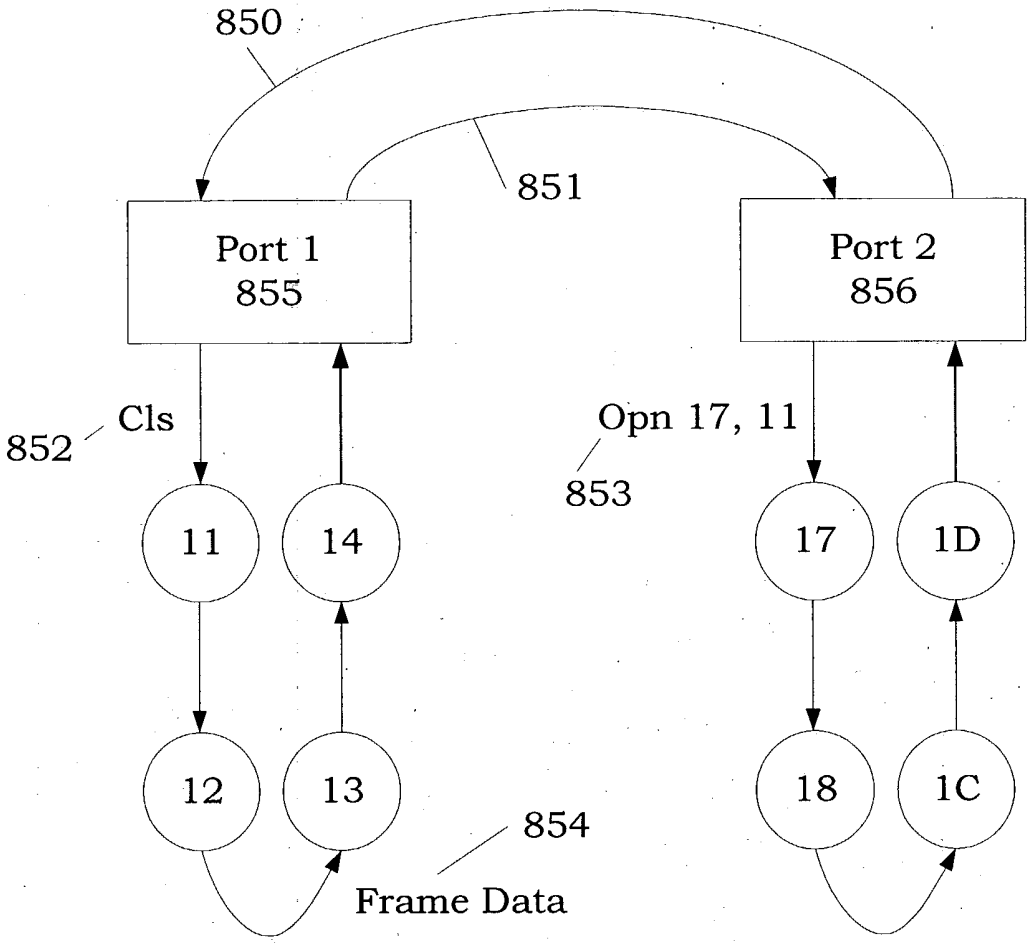
Close source by node 13.
Forwarded by node 12.
Arb ff Detected on port rx_data.
Disconnect.
Source idles.

Fig 11d



Arbs forwarded.
Open sourced.

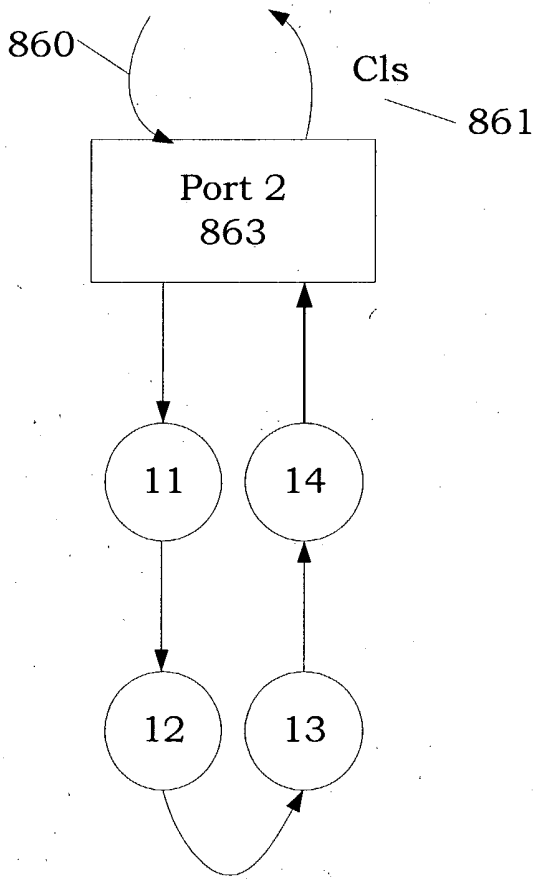
Fig 12a



Open forwarded to Port 2.
Data transferred.
Close sourced.

Open received from Port 1.
Data received.
Close forwarded.

Fig 12b



Close detected at transmillter and receiver. Connection broken.

Fig 12c

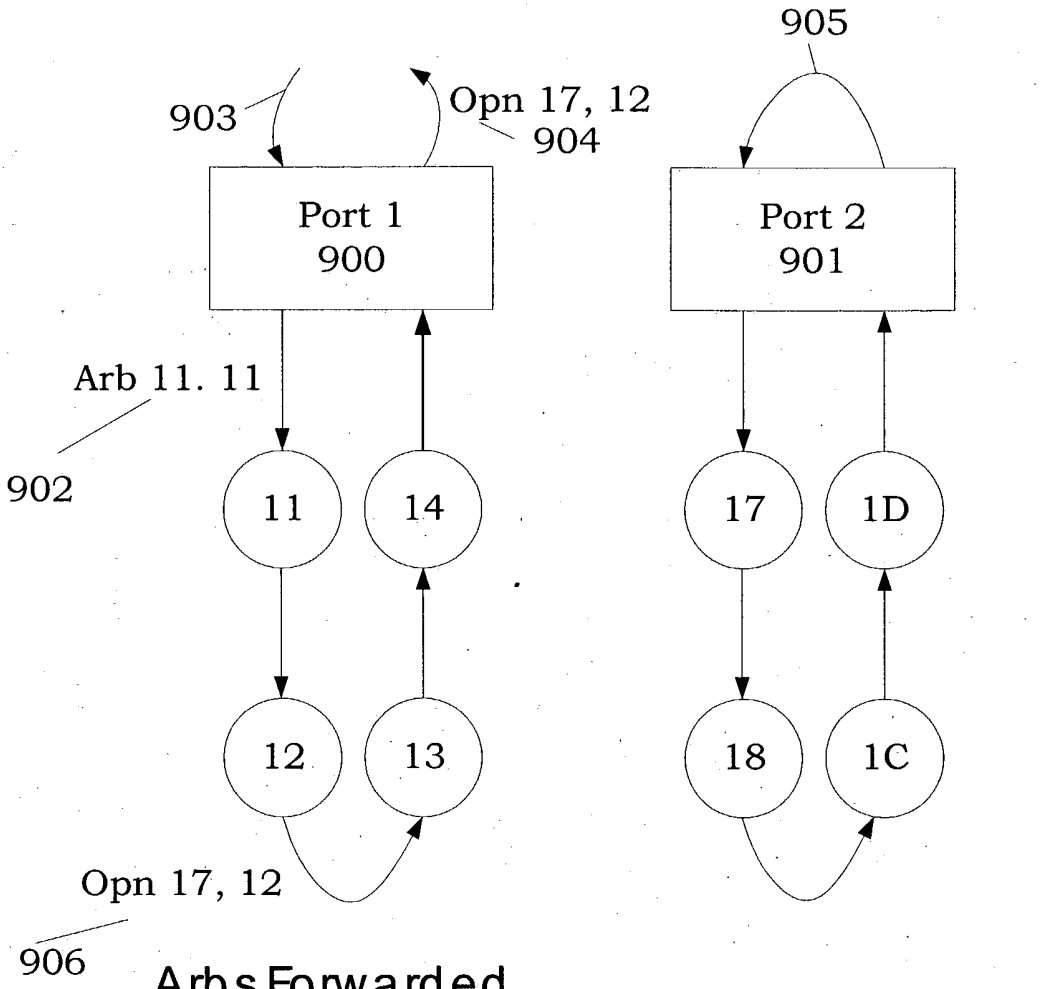
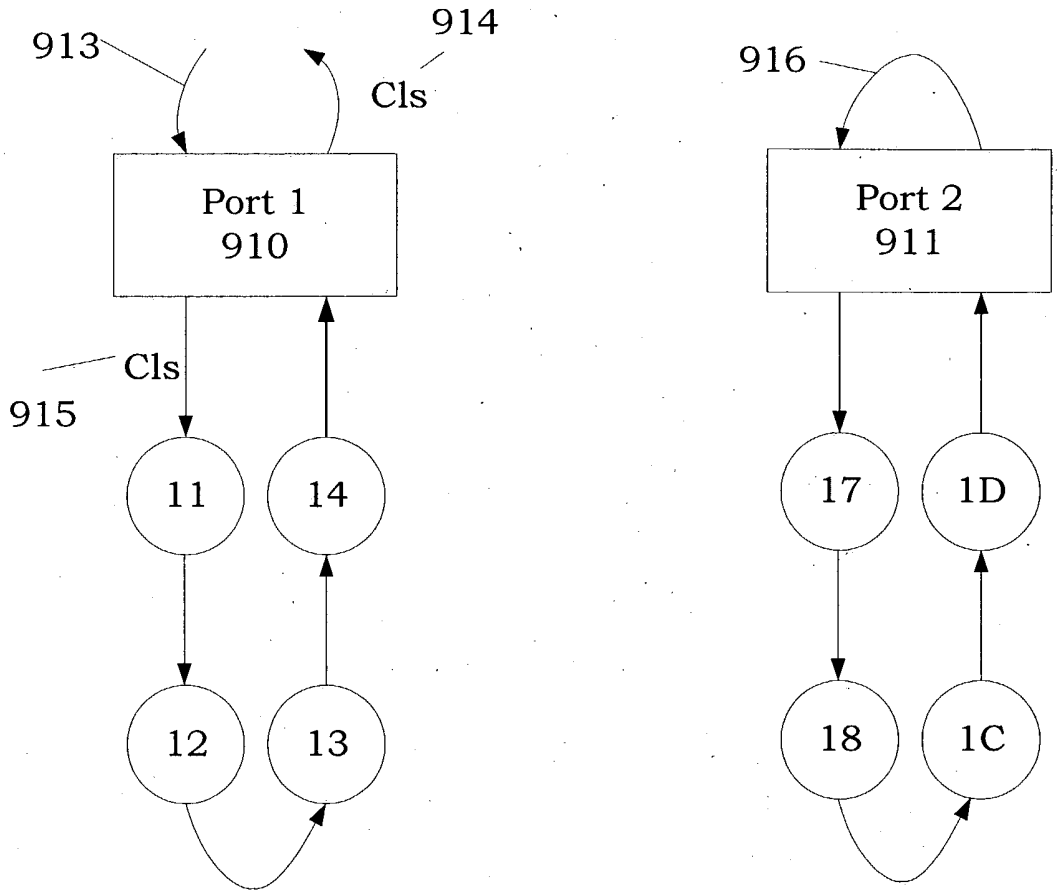

905
903
Opn 17, 12
904
Port 1
900
Port 2
901
Arb 11. 11
902
11
14
17
1D
12
13
18
1C
Opn 17, 12
906
Arbs Forwarded.
Open Sourced.
Port 2 Connected.

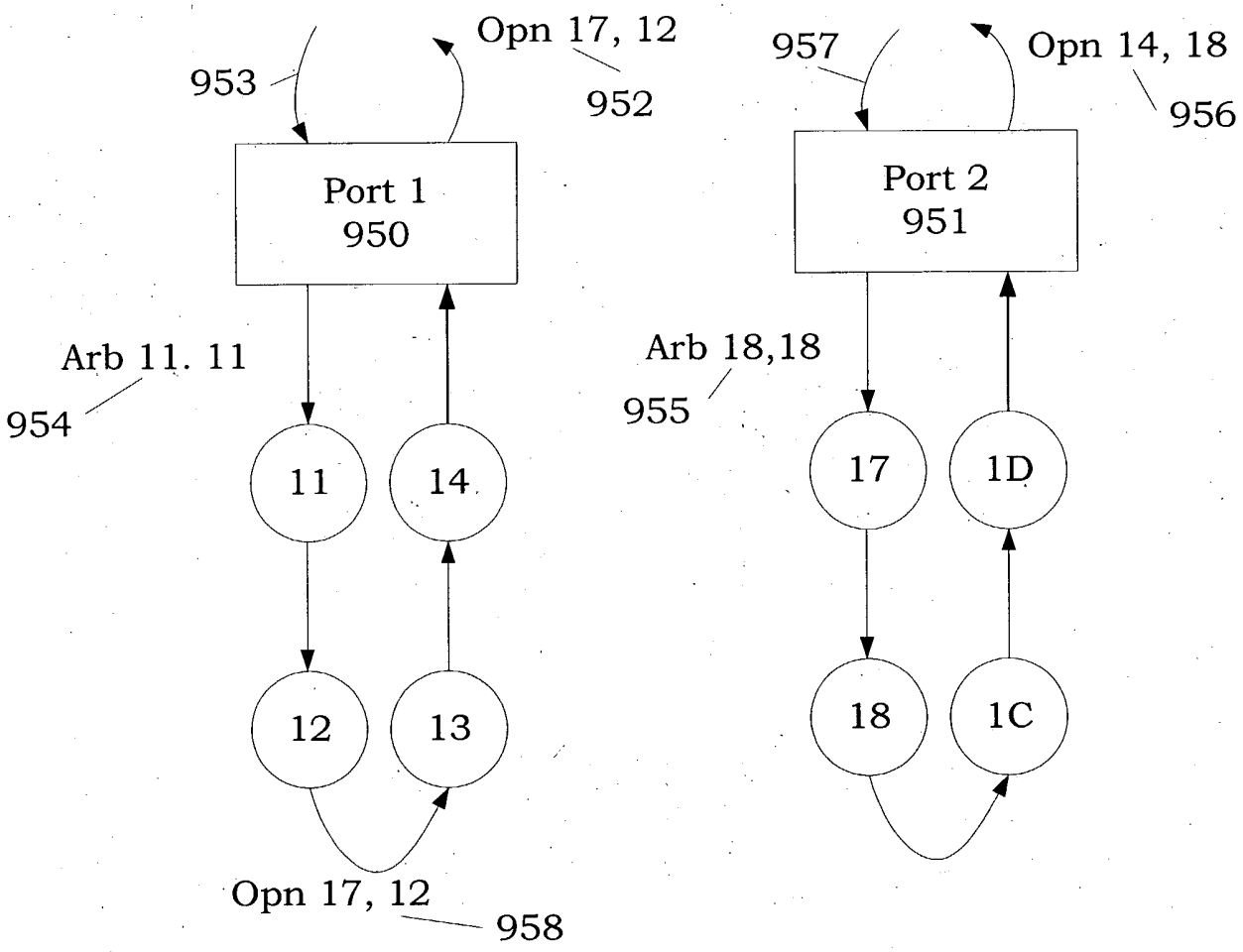
Fig 13a



Clsdetected at port receiver from node 12. Clssourced from port transmitter.

Port 2 still connected.

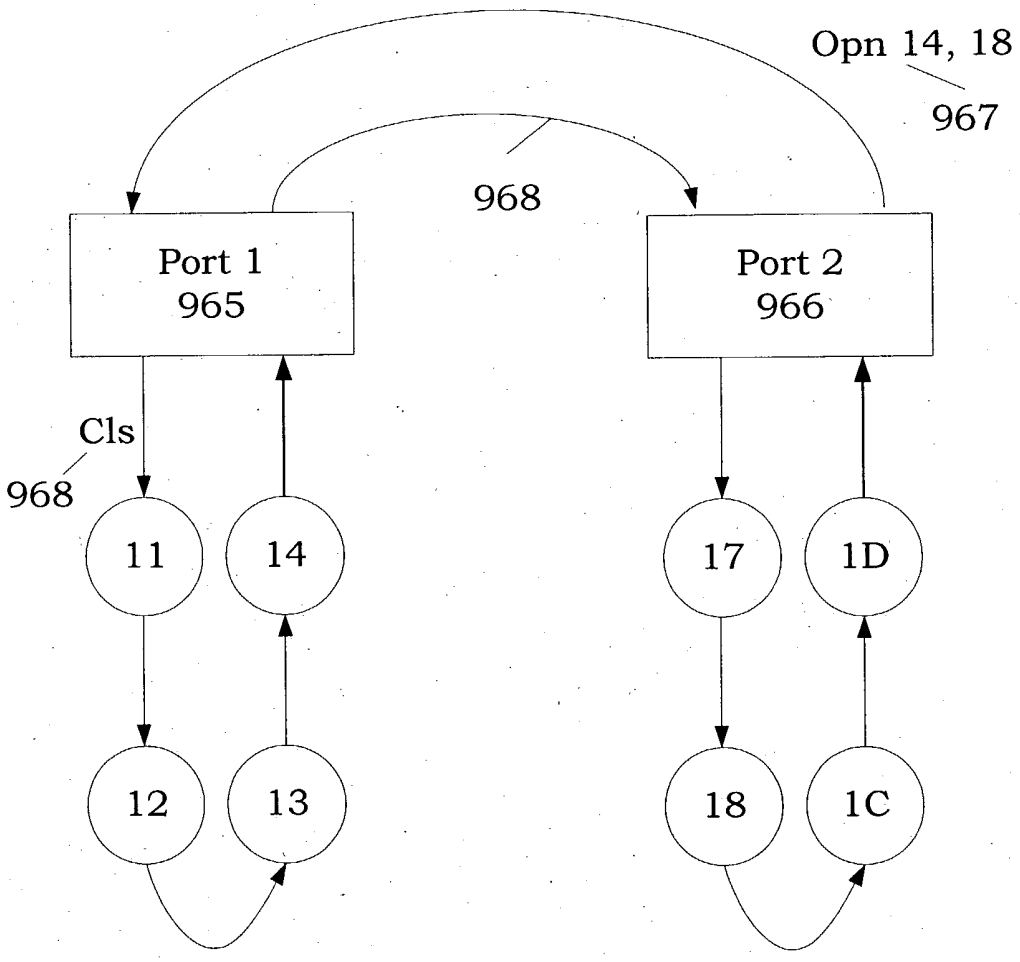
Fig 13b



Arbs forwarded.
Open sourced.

Arbs forwarded.
Open sourced.

Fig 14a

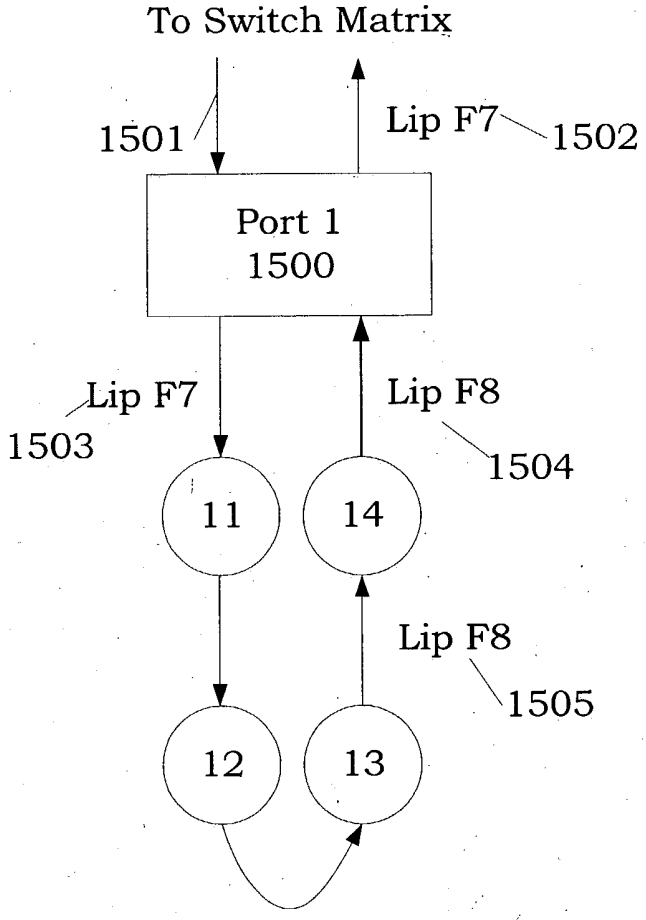


Port 1 connected as destination port. Cls sourced from Port 1 transmitter.

Port 2 connected as source Port. Open sourced from Port 2 receiver.

Fig 14b

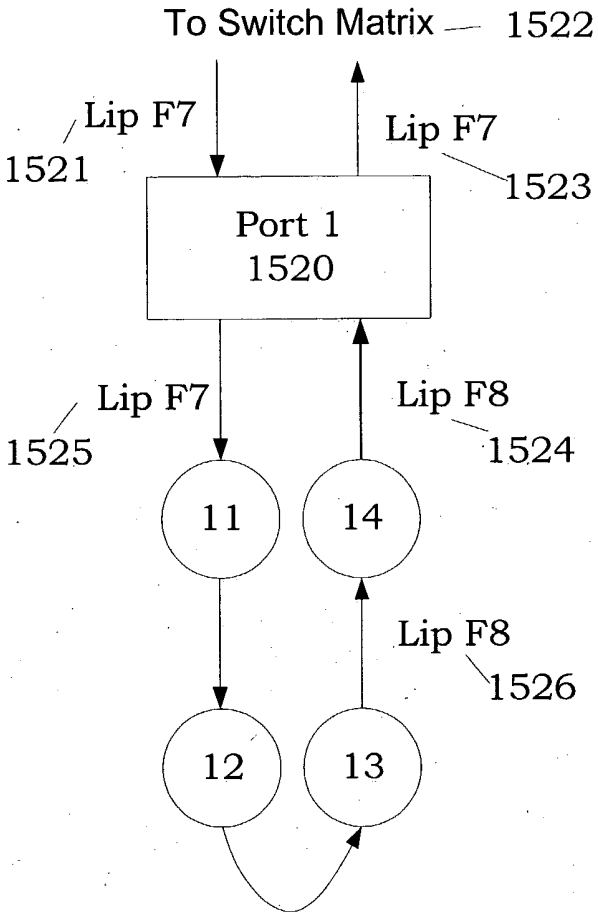
Router detects Lip RX'd signal.
Router deasserts switched mode signal.



Lip F8 sourced by node 13.
Lip F8 replaced by Lip F7 at port receiver.
Lip F7 sourced at port transmitter.

Fig 15a

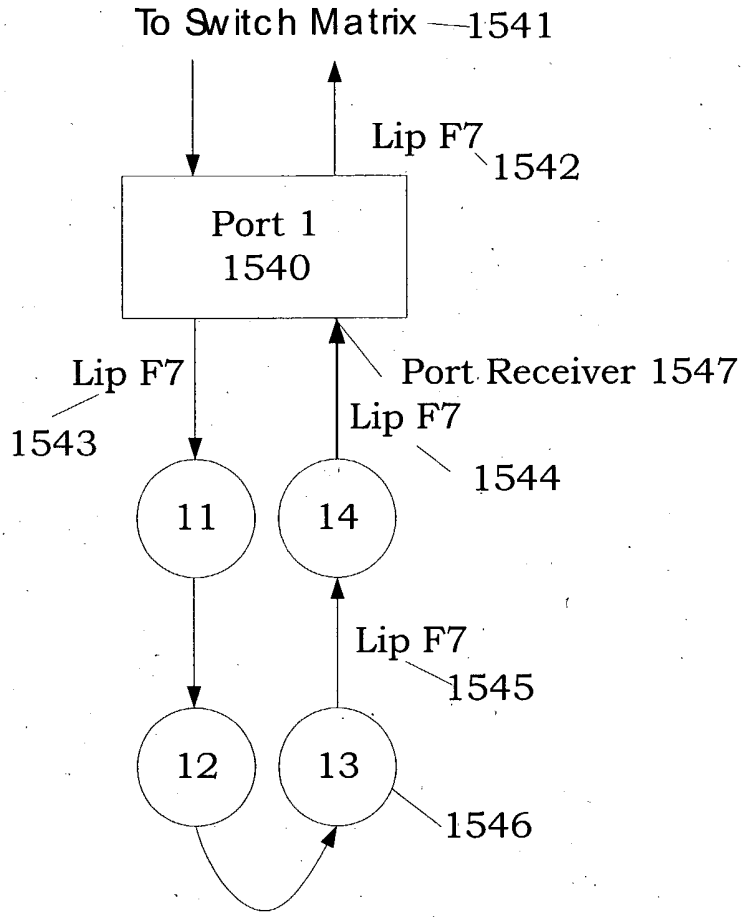
Router detects deassertion of port active.
Router takes port out of operational loop.



Lip F7 received at port transmitter.
Port deasserts port active signal.

Fig 15b

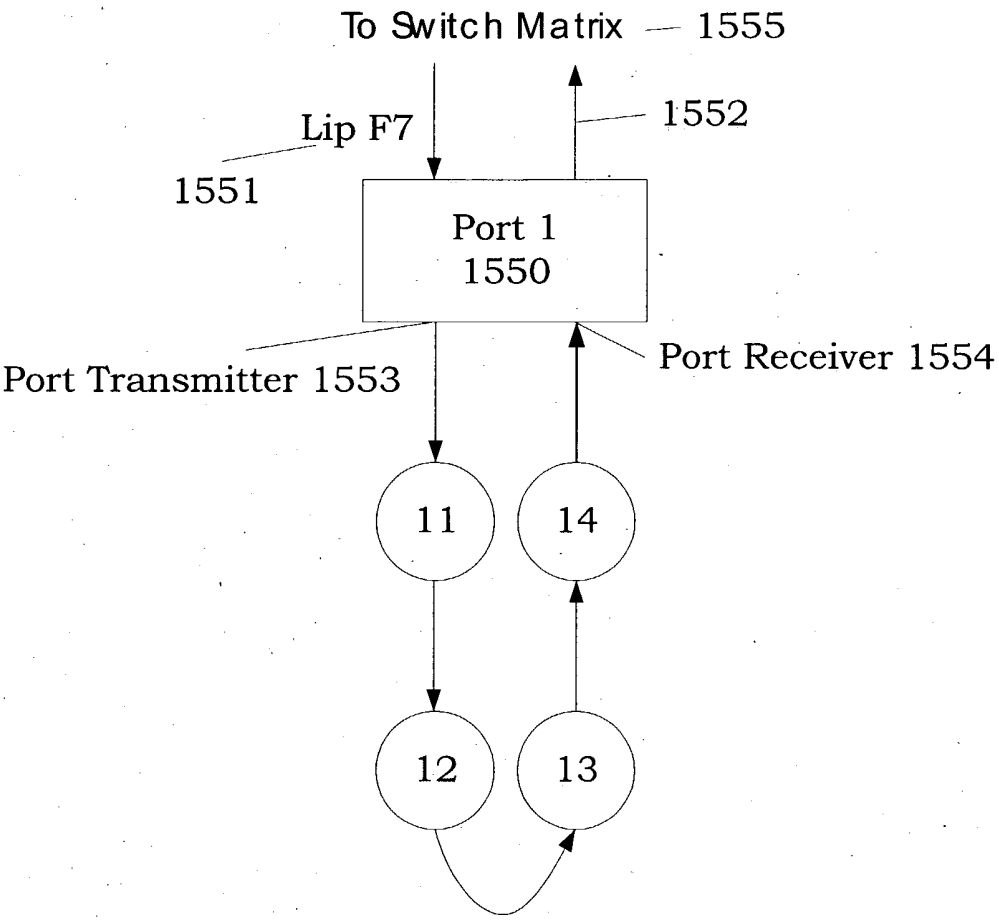
Router detects Lip RX'd signal since port active is now asserted.
Router puts port back into the operaional loop.



Node 13 "heals", source Lip F7.
Lip F7 received at port receiver.
Port asserts Port_Active.

Fig 15c

Router continues with loop initialization



Port detects Lip F7 at port transmitter.
Port stops sourcing Lip F7 at port receiver.
Port stops sourcing Lip F7 at port transmitter.

Fig 15d

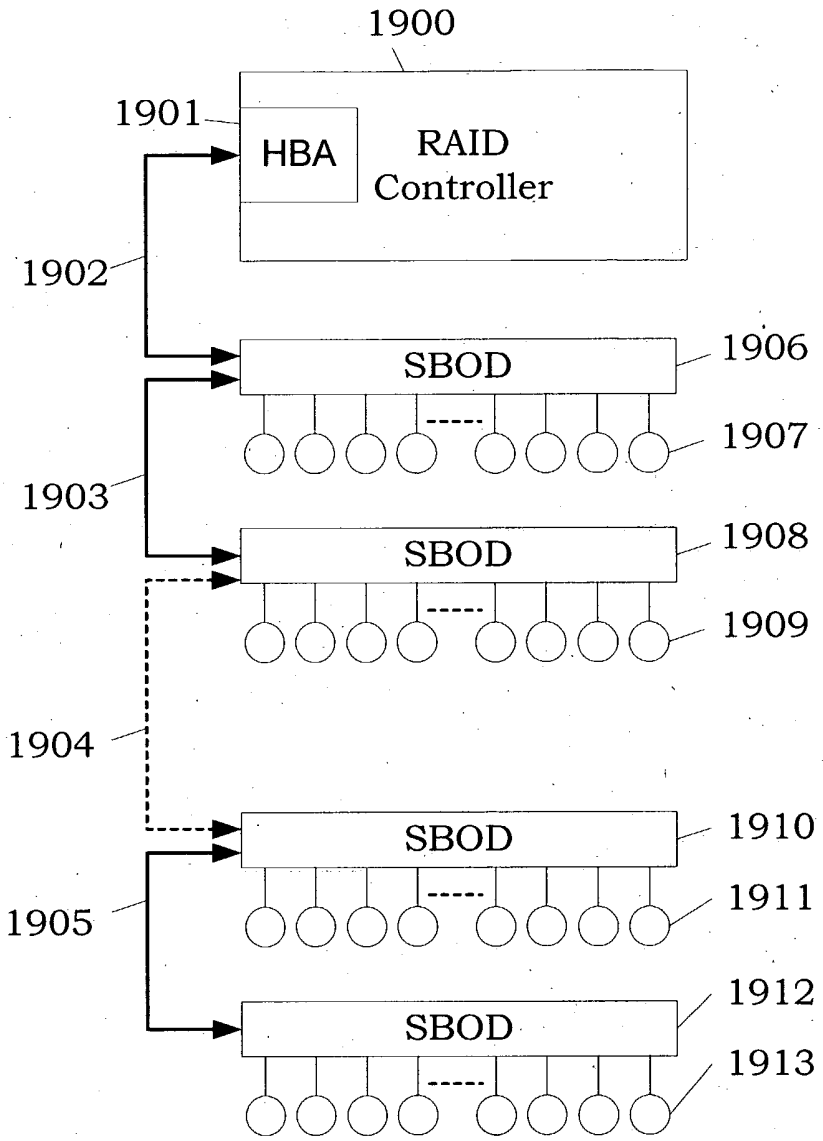

1900
1901
HBA
RAID Controller
1902
SBOD
1906
1907
1903
SBOD
1908
1909
1904
SBOD
1910
1911
1905
SBOD
1912
1913

Fig. 16a

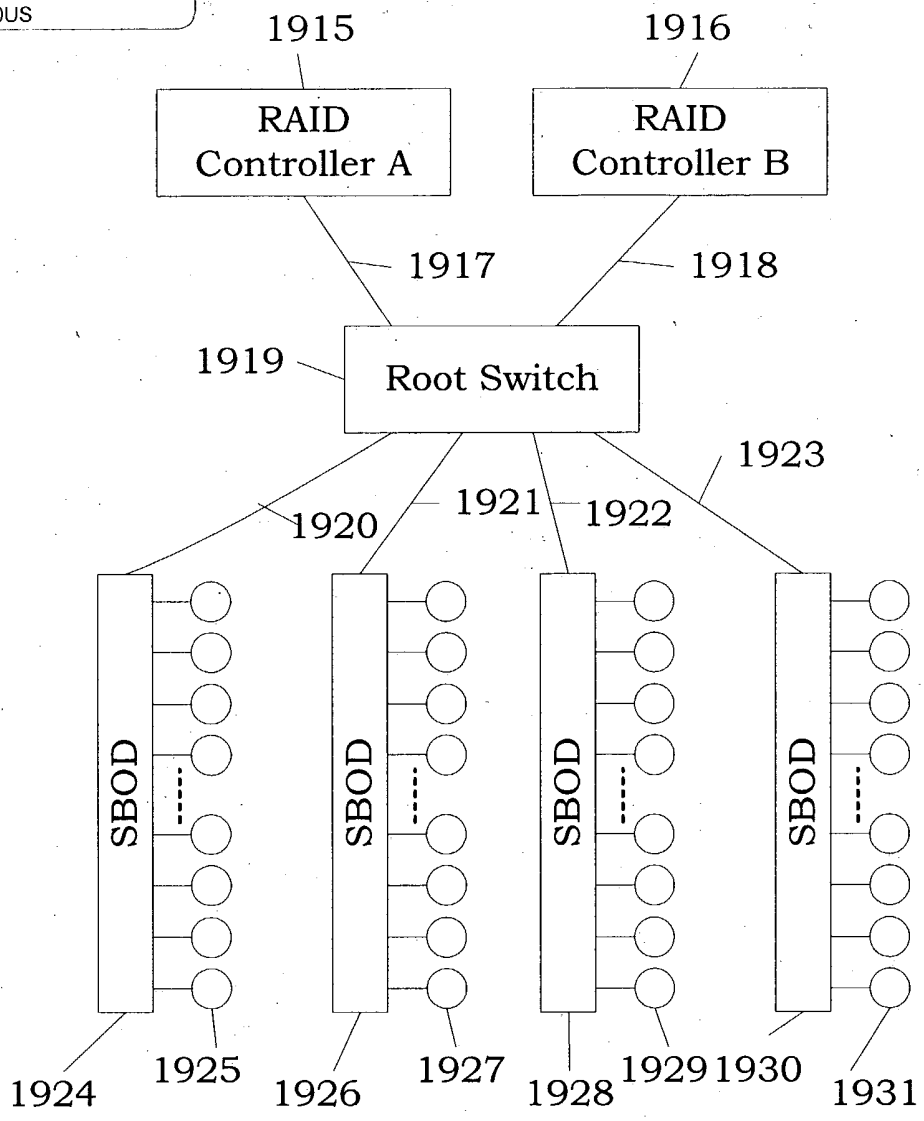

1915
1916
RAID Controller A
RAID Controller B
1917
1918
1919
Root Switch
1923
1920
1921
1922
SBOD
SBOD
SBOD
SBOD
1924
1925
1926
1927
1928
1929
1930
1931

Fig. 16b

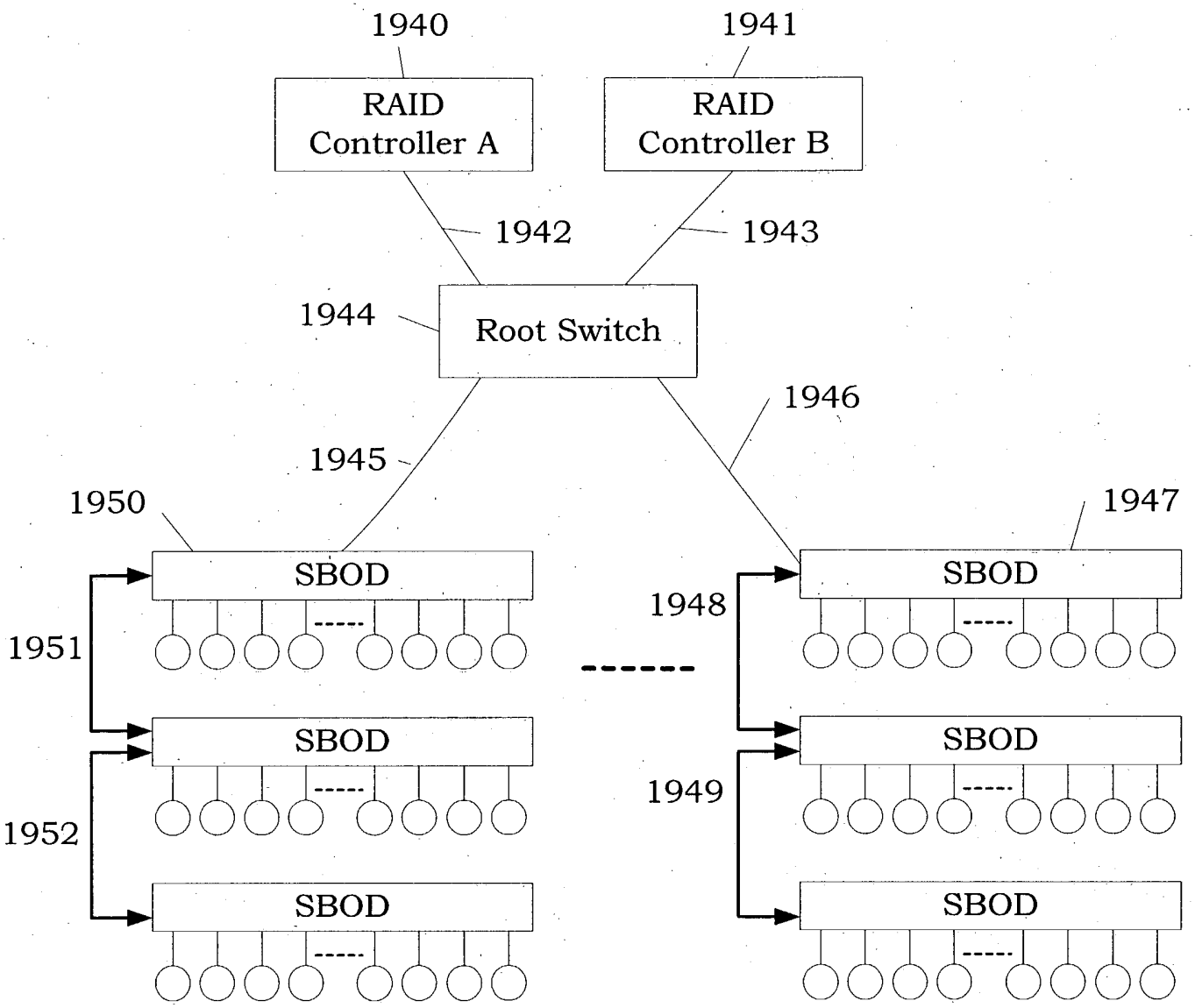

1940
1941
RAID Controller A
RAID Controller B
1942
1943
1944
Root Switch
1946
1945
1950
1947
SBOD
1948
1951
SBOD
SBOD
1949
1952
SBOD
SBOD

Fig. 16c

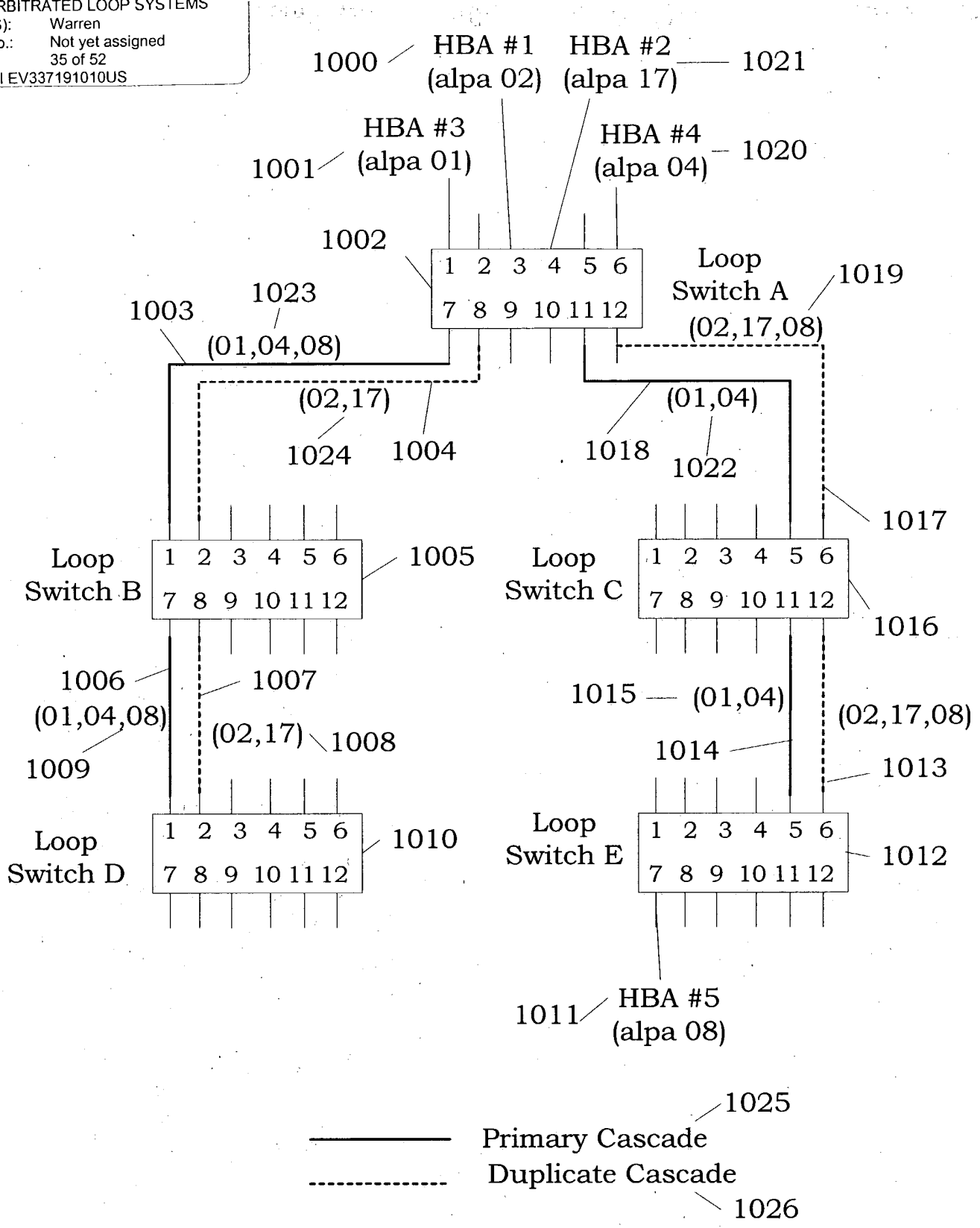

1000
HBA #1
(alpa 02)
HBA #2
(alpa 17)
1021
HBA #3
(alpa 01)
1001
HBA #4
(alpa 04)
1020
1002
1 2 3 4 5 6
7 8 9 10 11 12
Loop
Switch A
1019
(02,17,08)
1003
1023
(01,04,08)
(02,17)
1024
1004
1018
(01,04)
1022
1017
Loop
Switch B
1005
Loop
Switch C
1016
1006
(01,04,08)
1009
1007
(02,17)
1008
1015
(01,04)
(02,17,08)
1014
1013
Loop
Switch D
1010
Loop
Switch E
1012
1011
HBA #5
(alpa 08)
1025
Primary Cascade
Duplicate Cascade
1026

Fig. 17

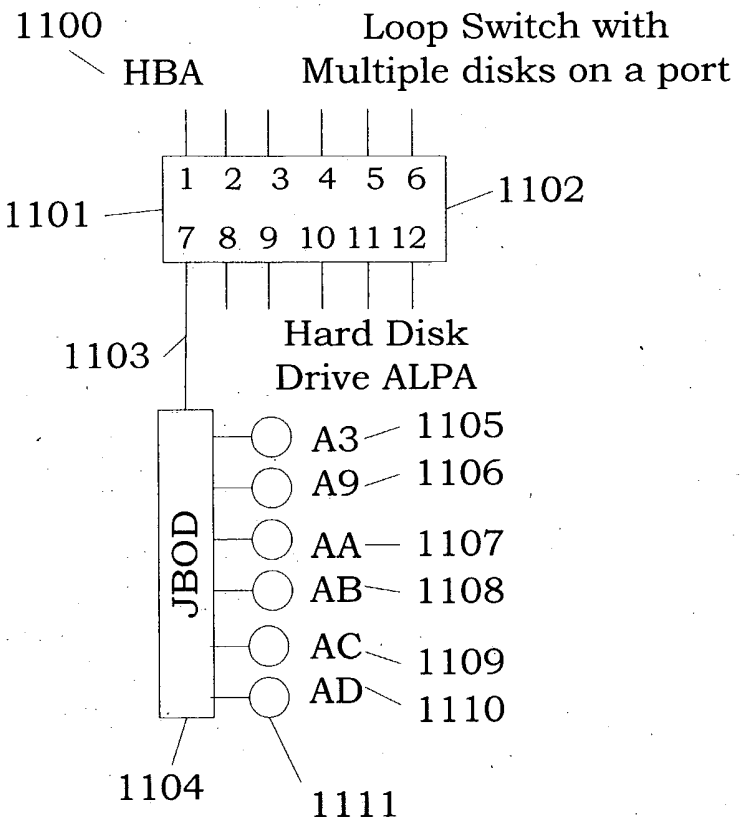
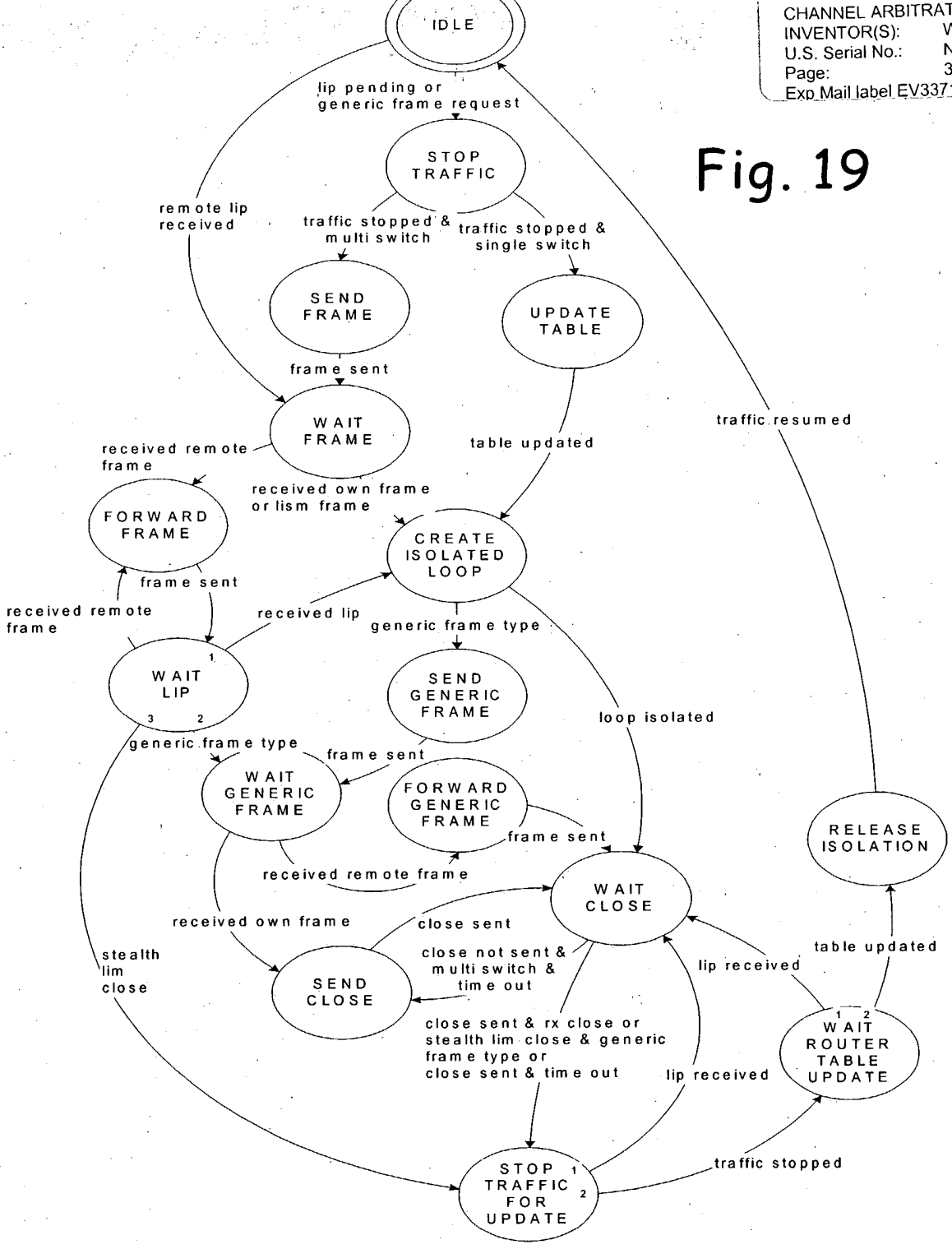

1100
HBA
Loop Switch with
Multiple disks on a port
1 2 3 4 5 6
7 8 9 10 11 12
1101
1102
1103
Hard Disk
Drive ALPA
JBOD
A3
A9
AA
AB
AC
AD
1105
1106
1107
1108
1109
1110
1104
1111

Fig. 18

Fig. 19


IDLE
lip pending or generic frame request
STOP TRAFFIC
remote lip received
traffic stopped & multi switch
traffic stopped & single switch
SEND FRAME
UPDATE TABLE
frame sent
WAIT FRAME
table updated
traffic resumed
received remote frame
received own frame or lism frame
FORWARD FRAME
CREATE ISOLATED LOOP
frame sent
received remote frame
received lip
generic frame type
WAIT LIP
SEND GENERIC FRAME
loop isolated
generic frame type
frame sent
WAIT GENERIC FRAME
FORWARD GENERIC FRAME
frame sent
RELEASE ISOLATION
received remote frame
WAIT CLOSE
received own frame
close sent
table updated
stealth lim close
close not sent & multi switch & time out
lip received
SEND CLOSE
close sent & rx close or stealth lim close & generic frame type or close sent & time out
WAIT ROUTER TABLE UPDATE
lip received
STOP TRAFFIC FOR UPDATE
traffic stopped

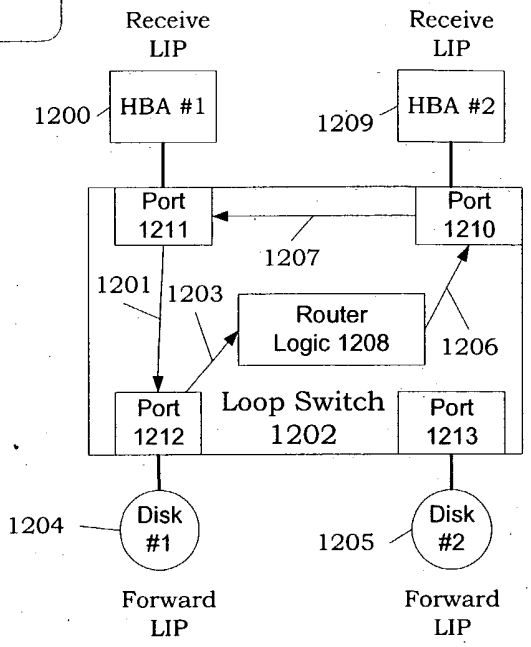

Receive LIP
Receive LIP
1200
HBA #1
1209
HBA #2
Port 1211
Port 1210
1207
1201
1203
Router Logic 1208
1206
Port 1212
Loop Switch 1202
Port 1213
1204
Disk #1
1205
Disk #2
Forward LIP
Forward LIP

Fig. 20

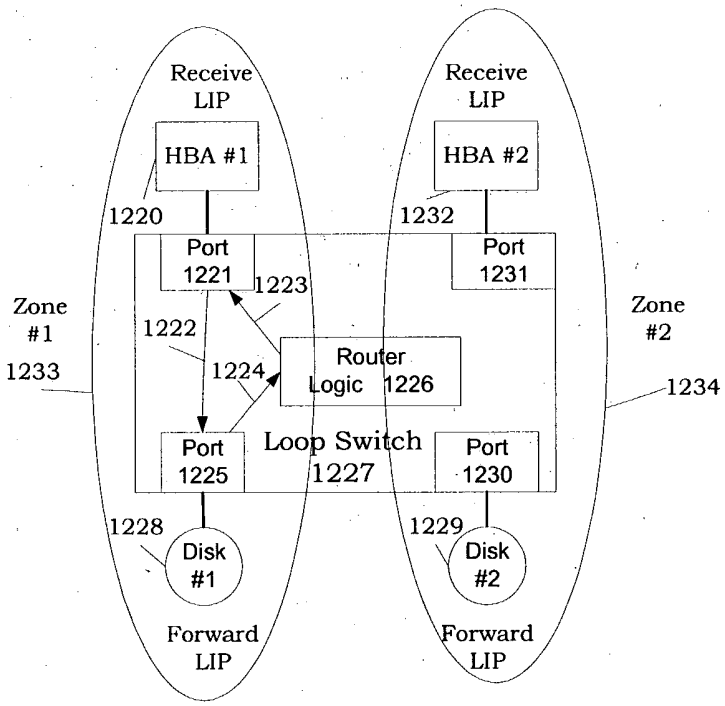

Receive LIP
Receive LIP
HBA #1
HBA #2
1220
1232
Port 1221
Port 1231
1223
Zone #1
1222
Zone #2
1233
1224
Router Logic 1226
1234
Port 1225
Loop Switch 1227
Port 1230
1228
1229
Disk #1
Disk #2
Forward LIP
Forward LIP

Fig. 21

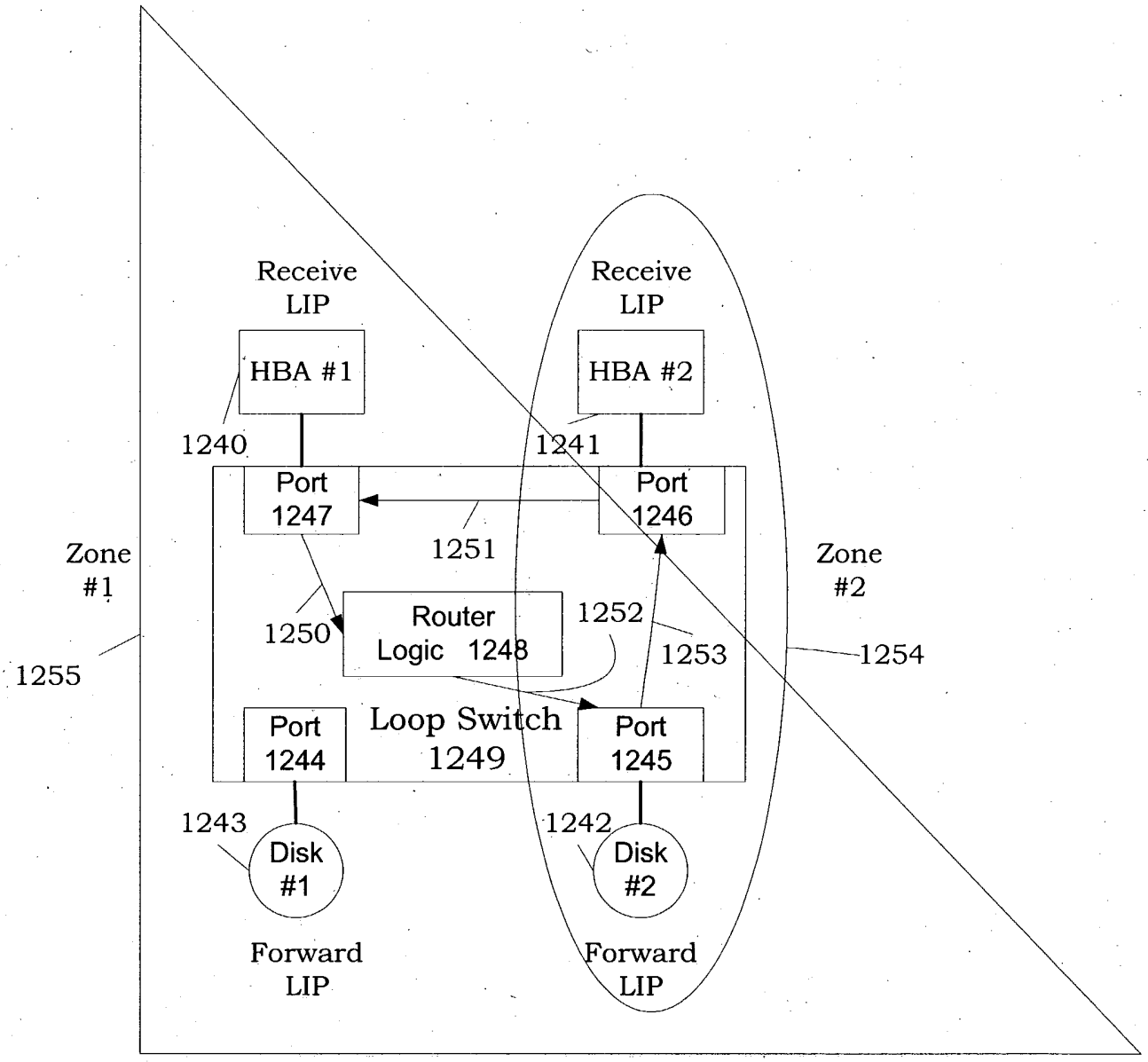

Receive LIP
Receive LIP
HBA #1
HBA #2
1240
1241
Port 1247
Port 1246
1251
Zone #1
Zone #2
1250
Router Logic 1248
1252
1253
1254
1255
Port 1244
Loop Switch 1249
Port 1245
1243
1242
Disk #1
Disk #2
Forward LIP
Forward LIP

Fig. 22

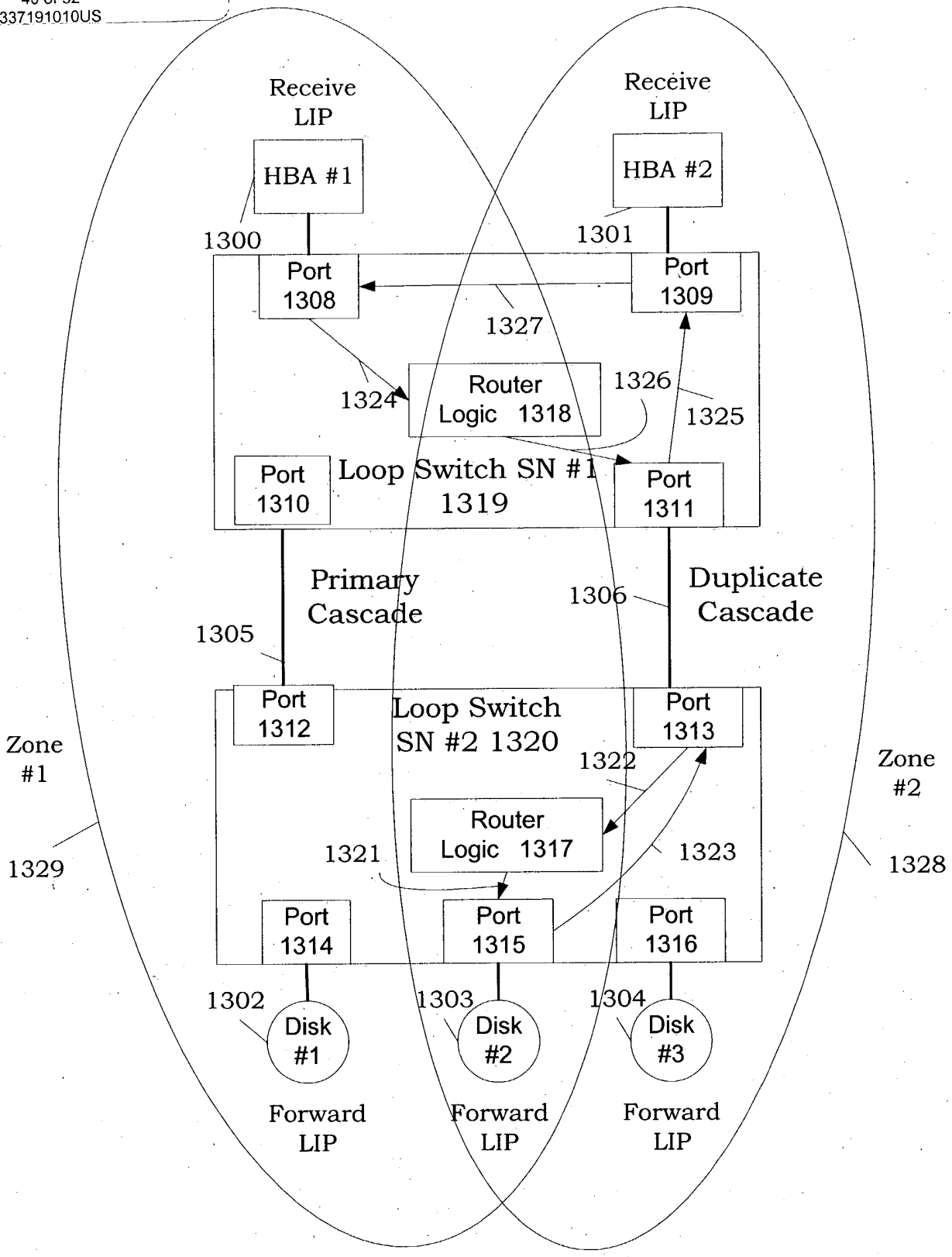

Receive LIP
HBA #1
1300
Receive LIP
HBA #2
1301
Port 1308
Port 1309
1327
1324
Router Logic 1318
1326
1325
Port 1310
Loop Switch SN #1 1319
Port 1311
Primary Cascade
1305
1306
Duplicate Cascade
Port 1312
Loop Switch SN #2 1320
Port 1313
1322
Zone #1
1329
Router Logic 1317
1321
1323
Zone #2
1328
Port 1314
Port 1315
Port 1316
1302
1303
1304
Disk #1
Disk #2
Disk #3
Forward LIP
Forward LIP
Forward LIP

Fig. 23

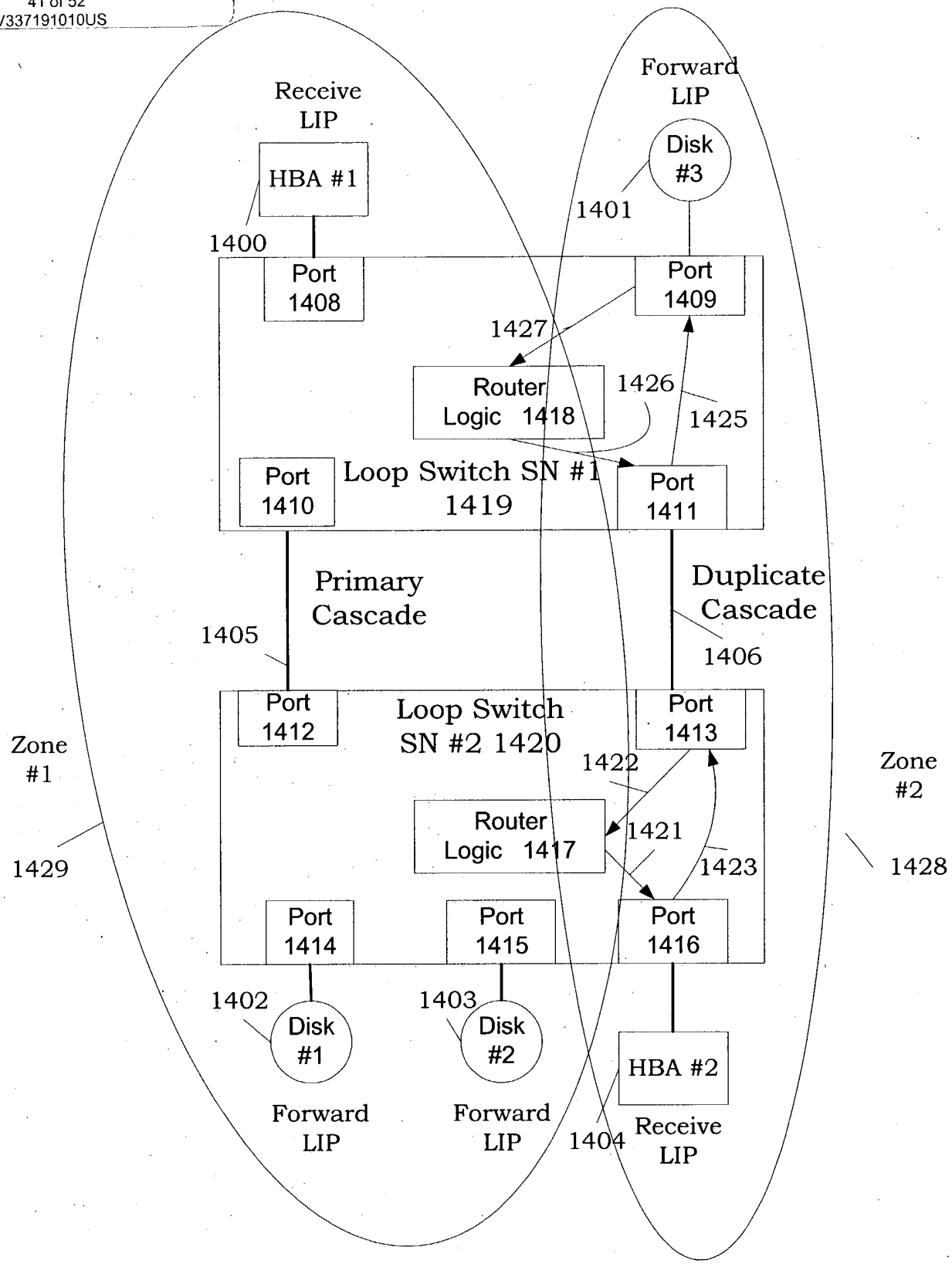

Receive LIP
Forward LIP
HBA #1
Disk #3
1401
1400
Port 1408
Port 1409
1427
Router Logic 1418
1426
1425
Port 1410
Loop Switch SN #1 1419
Port 1411
Primary Cascade
Duplicate Cascade
1405
1406
Port 1412
Loop Switch SN #2 1420
Port 1413
1422
Zone #1
Zone #2
Router Logic 1417
1421
1423
1429
1428
Port 1414
Port 1415
Port 1416
1402
1403
Disk #1
Disk #2
HBA #2
Forward LIP
Forward LIP
1404
Receive LIP

Fig. 24

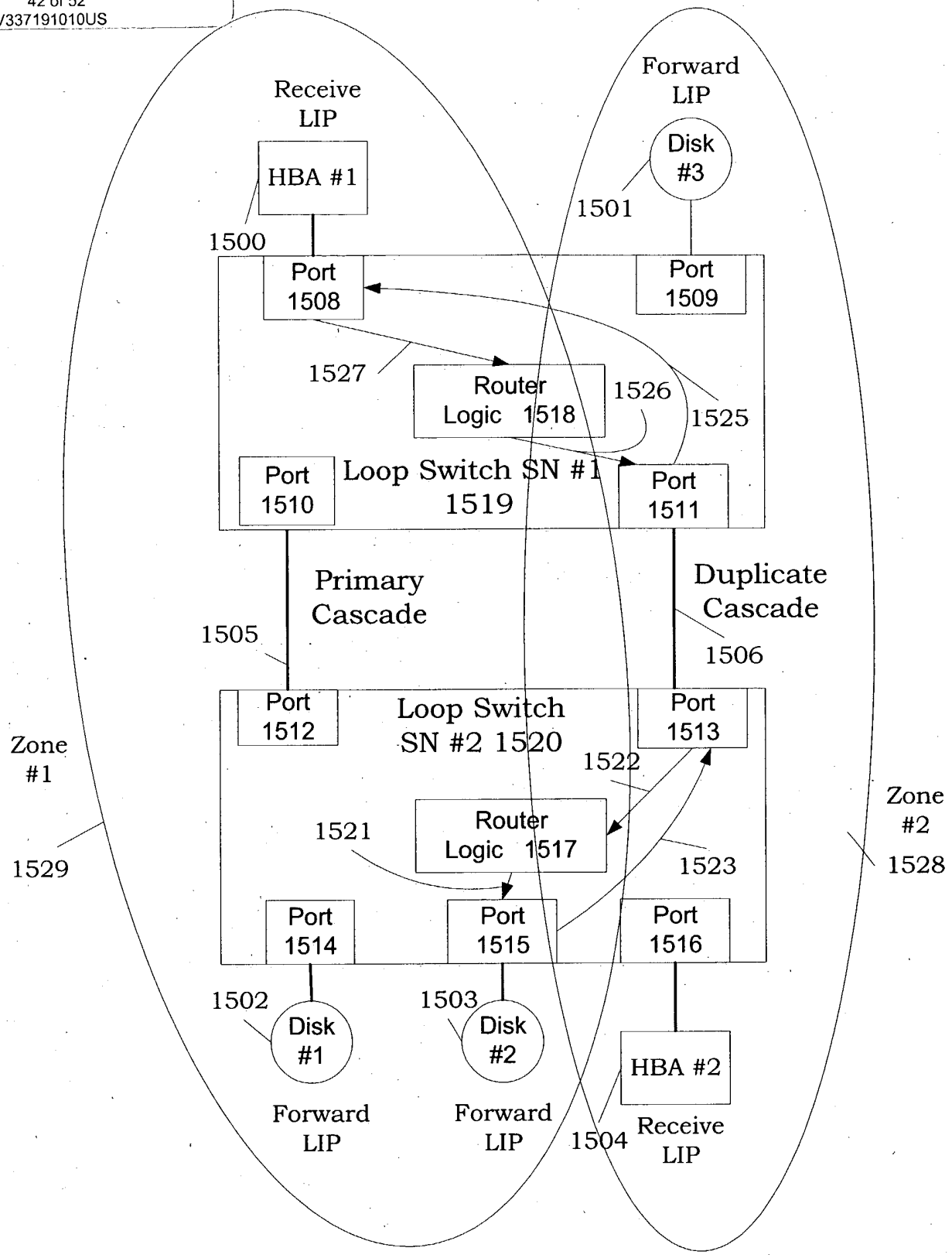

Receive LIP
Forward LIP
HBA #1
Disk #3
1501
1500
Port 1508
Port 1509
1527
Router Logic 1518
1526
1525
Port 1510
Loop Switch SN #1 1519
Port 1511
Primary Cascade
Duplicate Cascade
1505
1506
Port 1512
Loop Switch SN #2 1520
Port 1513
1522
Zone #1
1529
1521
Router Logic 1517
1523
Zone #2
1528
Port 1514
Port 1515
Port 1516
1502
1503
Disk #1
Disk #2
HBA #2
Forward LIP
Forward LIP
1504
Receive LIP

Fig. 25

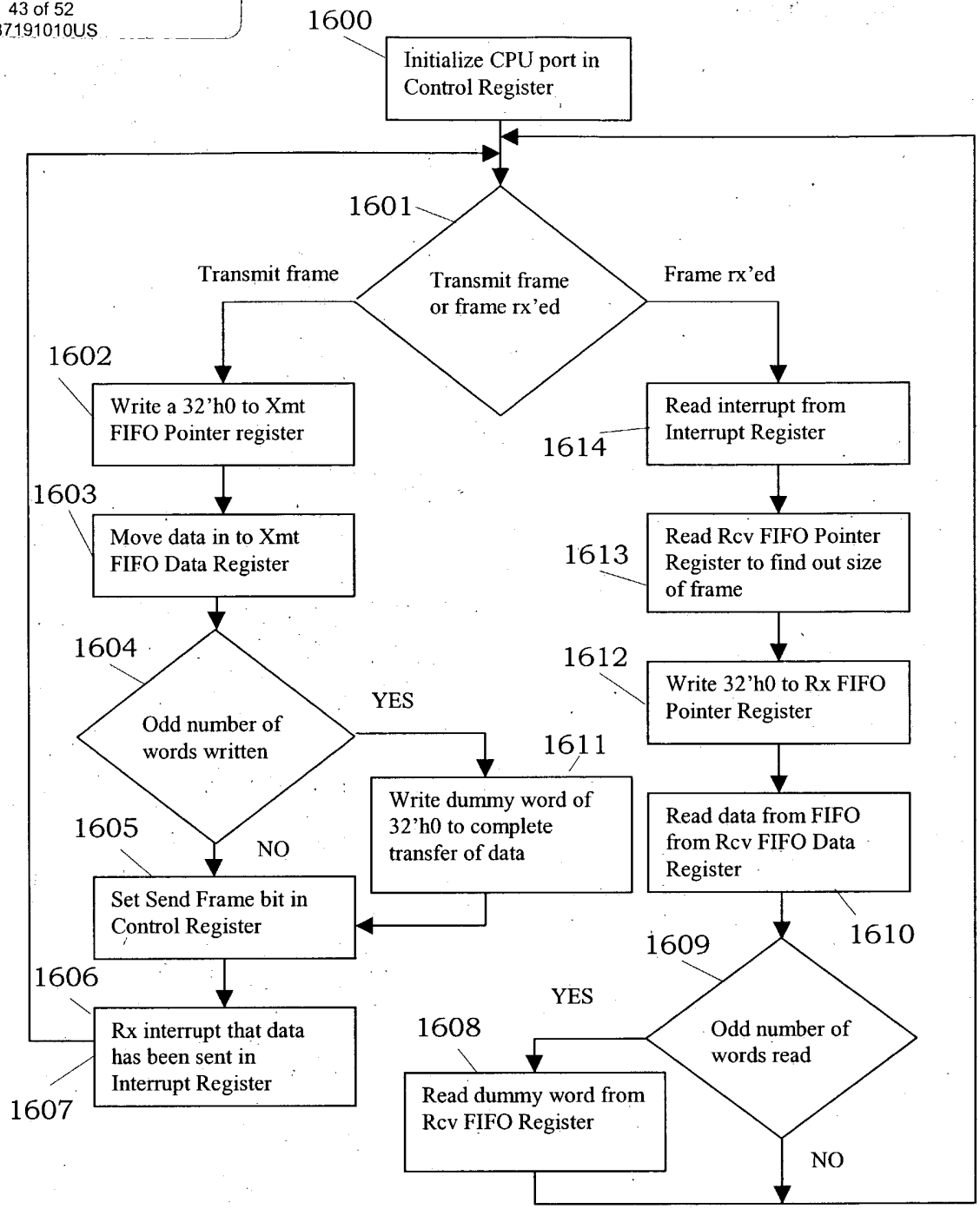

1600
Initialize CPU port in Control Register
1601
Transmit frame or frame rx'ed
Transmit frame
Frame rx'ed
1602
Write a 32'h0 to Xmt FIFO Pointer register
1603
Move data in to Xmt FIFO Data Register
1604
Odd number of words written
YES
NO
1605
Set Send Frame bit in Control Register
1606
Rx interrupt that data has been sent in Interrupt Register
1607
1611
Write dummy word of 32'h0 to complete transfer of data
1614
Read interrupt from Interrupt Register
1613
Read Rcv FIFO Pointer Register to find out size of frame
1612
Write 32'h0 to Rx FIFO Pointer Register
Read data from FIFO from Rcv FIFO Data Register
1610
1609
Odd number of words read
YES
NO
1608
Read dummy word from Rcv FIFO Register

Fig. 26

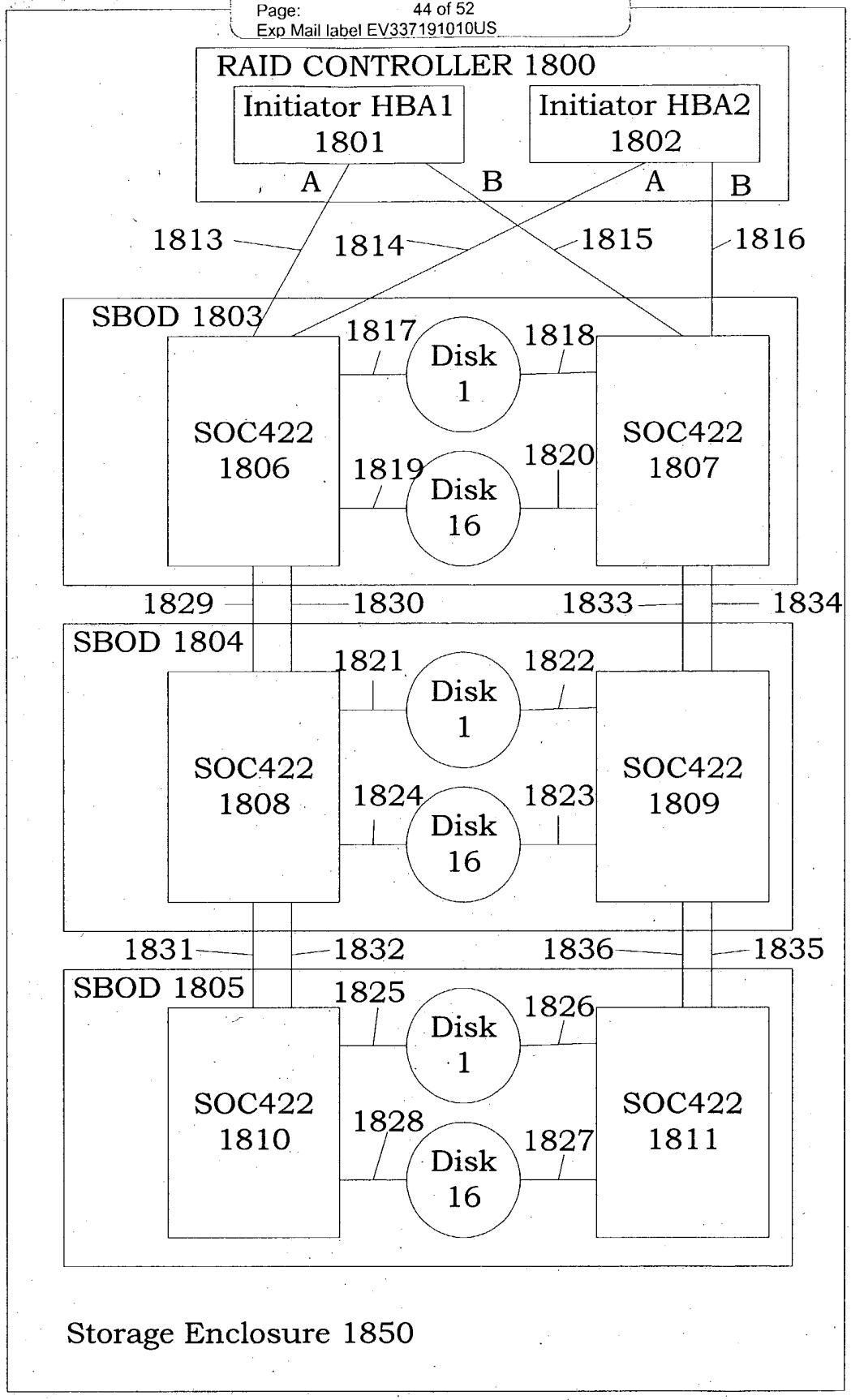
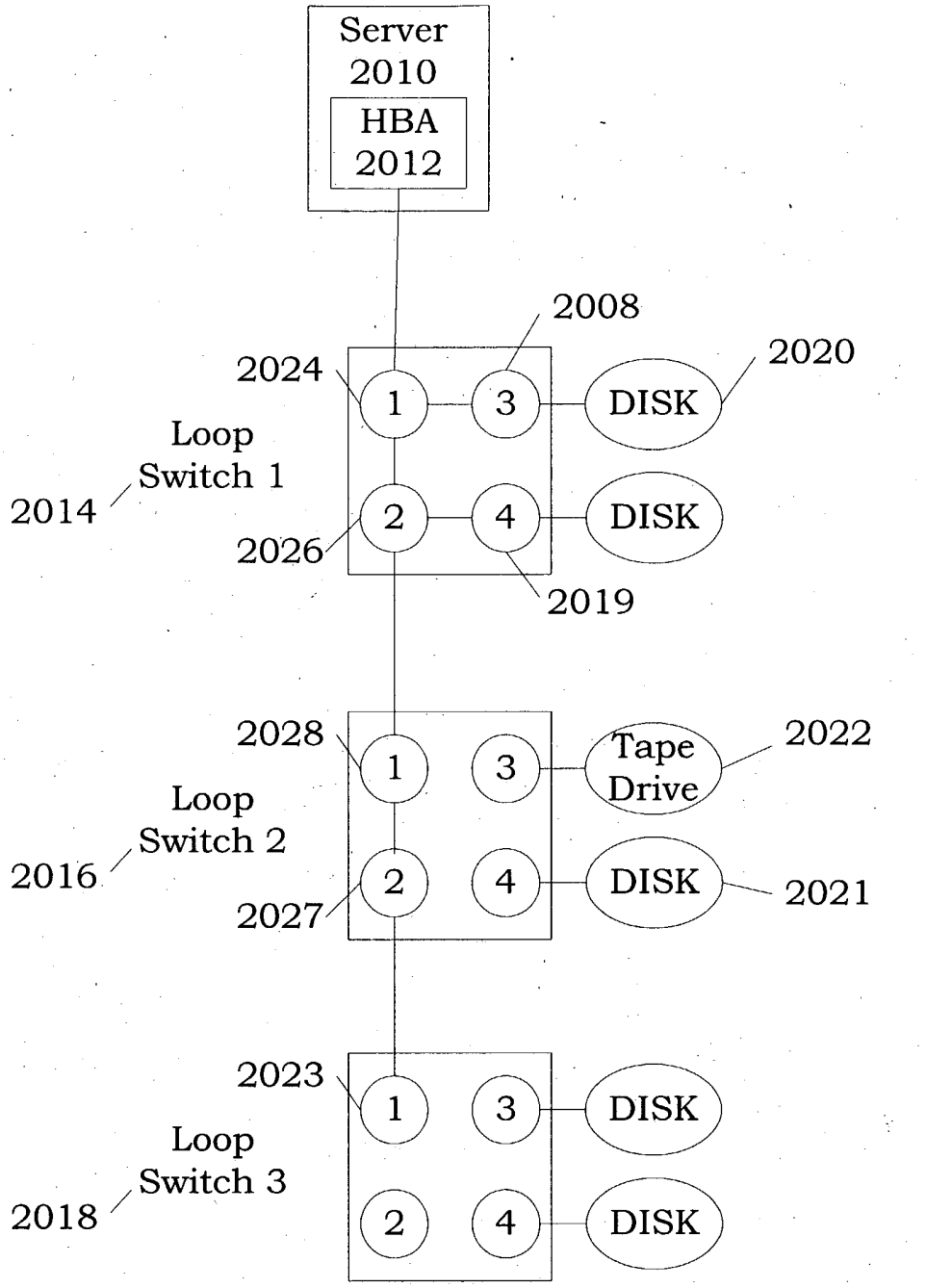
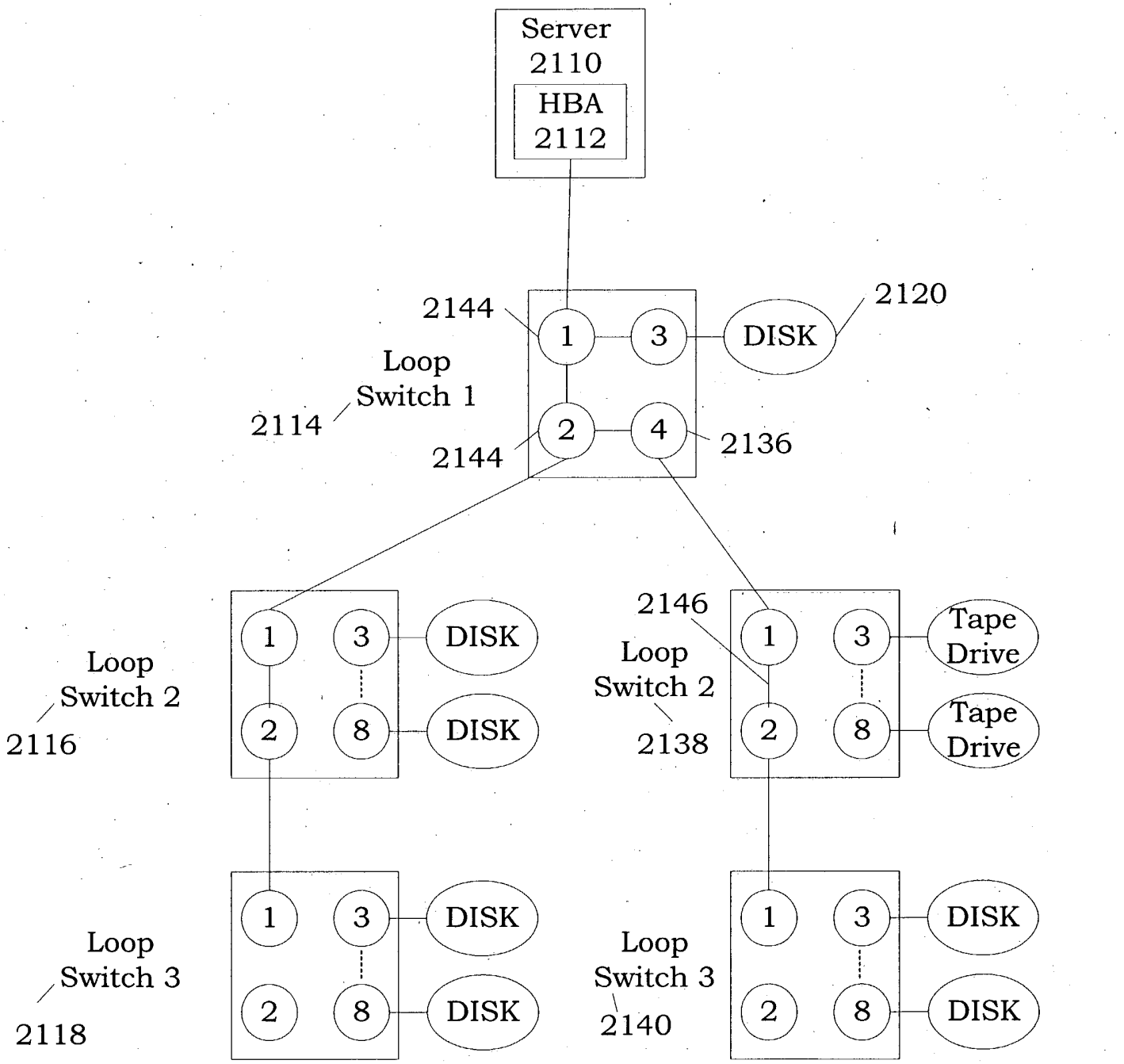

RAID CONTROLLER 1800
Initiator HBA1 1801
Initiator HBA2 1802
A
B
A
B
1813
1814
1815
1816
SBOD 1803
1817
Disk 1
1818
SOC422 1806
SOC422 1807
1819
Disk 16
1820
1829
1830
1833
1834
SBOD 1804
1821
Disk 1
1822
SOC422 1808
SOC422 1809
1824
Disk 16
1823
1831
1832
1836
1835
SBOD 1805
1825
Disk 1
1826
SOC422 1810
SOC422 1811
1828
Disk 16
1827
Storage Enclosure 1850

Fig. 27



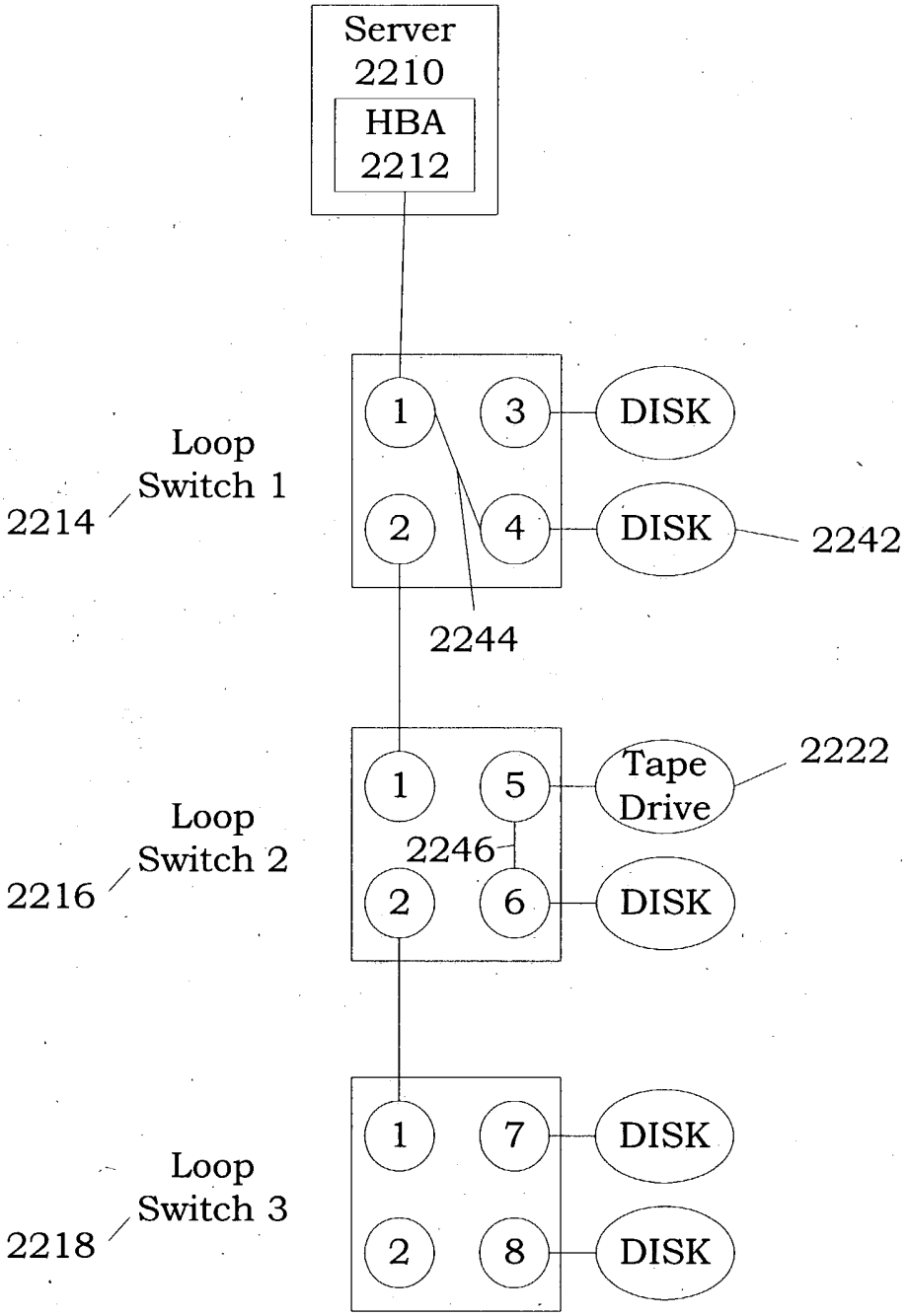
DISK = Fibre Channel
Hard Disk Drive

Fig. 28



DISK = Fibre Channel
Hard Disk Drive

Fig. 29



DISK = Fibre Channel
Hard Disk Drive

Fig. 30

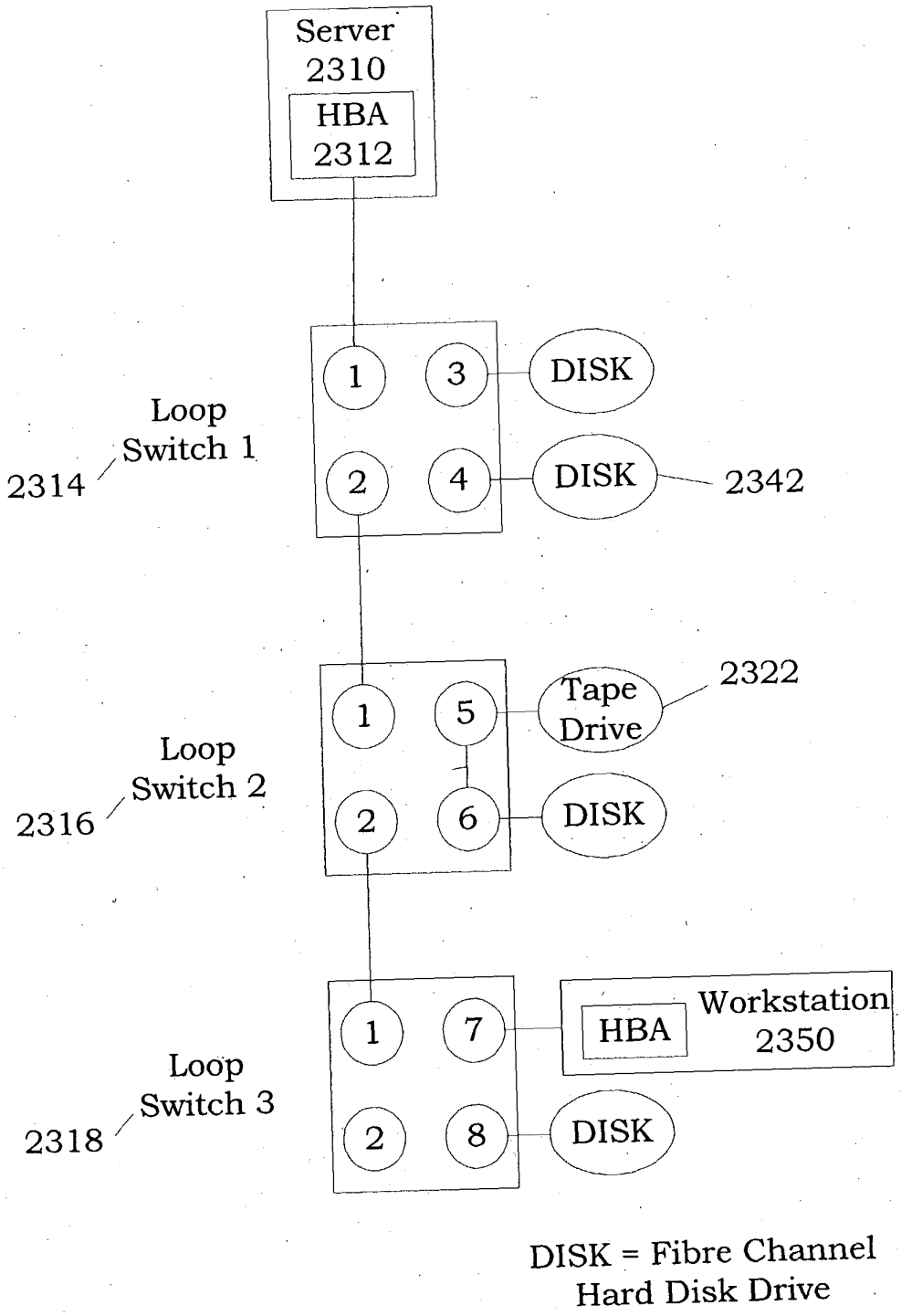

Server
2310
HBA
2312
Loop
Switch 1
2314
1
3
DISK
2
4
DISK
2342
Loop
Switch 2
2316
1
5
Tape
Drive
2322
2
6
DISK
Loop
Switch 3
2318
1
7
HBA
Workstation
2350
2
8
DISK
DISK = Fibre Channel
Hard Disk Drive

Fig. 31

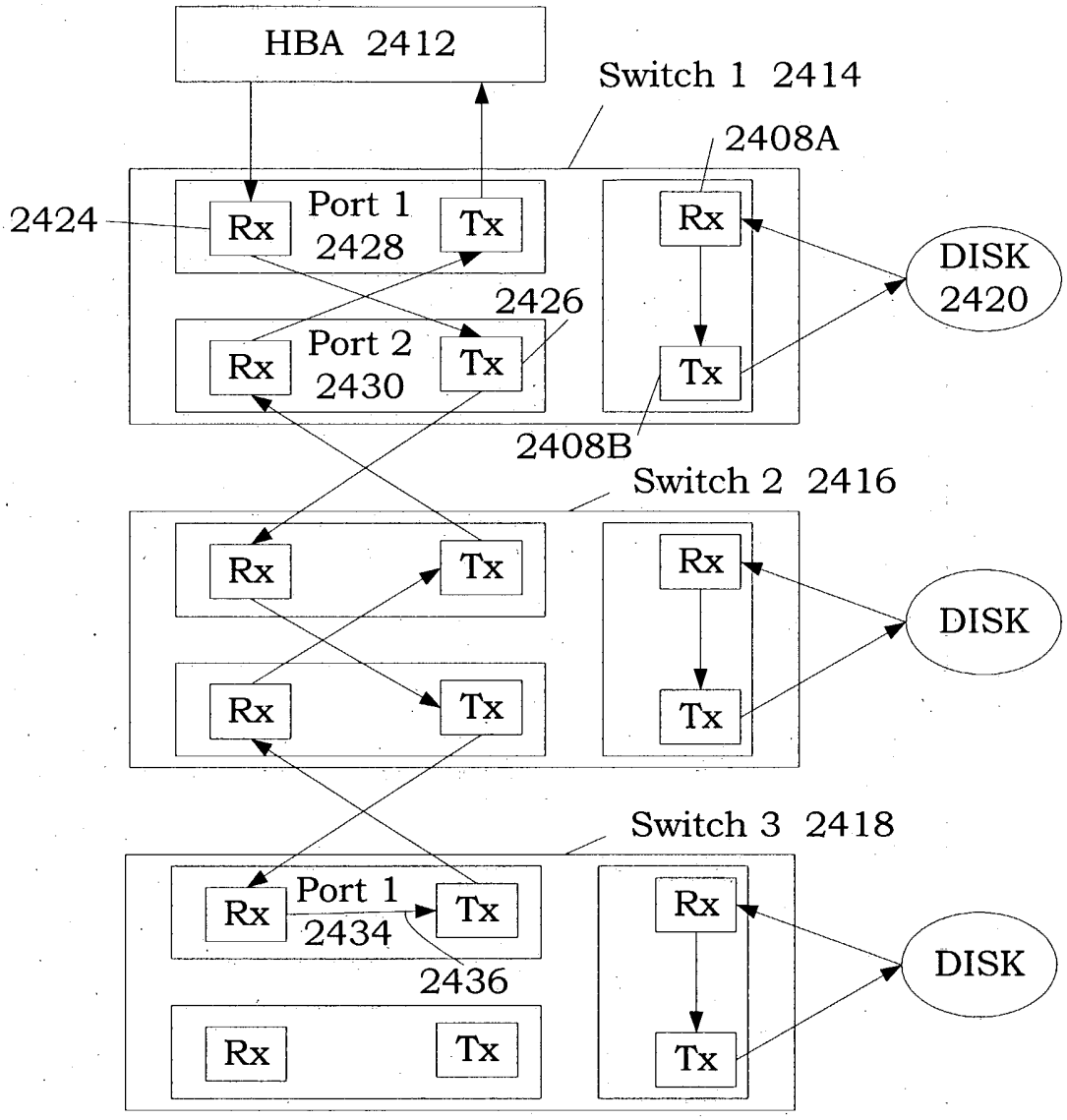

HBA 2412
Switch 1 2414
2408A
2424
Rx
Port 1
2428
Tx
Rx
DISK
2420
2426
Rx
Port 2
2430
Tx
Tx
2408B
Switch 2 2416
Rx
Tx
Rx
DISK
Rx
Tx
Tx
Switch 3 2418
Rx
Port 1
2434
Tx
Rx
DISK
2436
Rx
Tx
Tx

Fig. 32

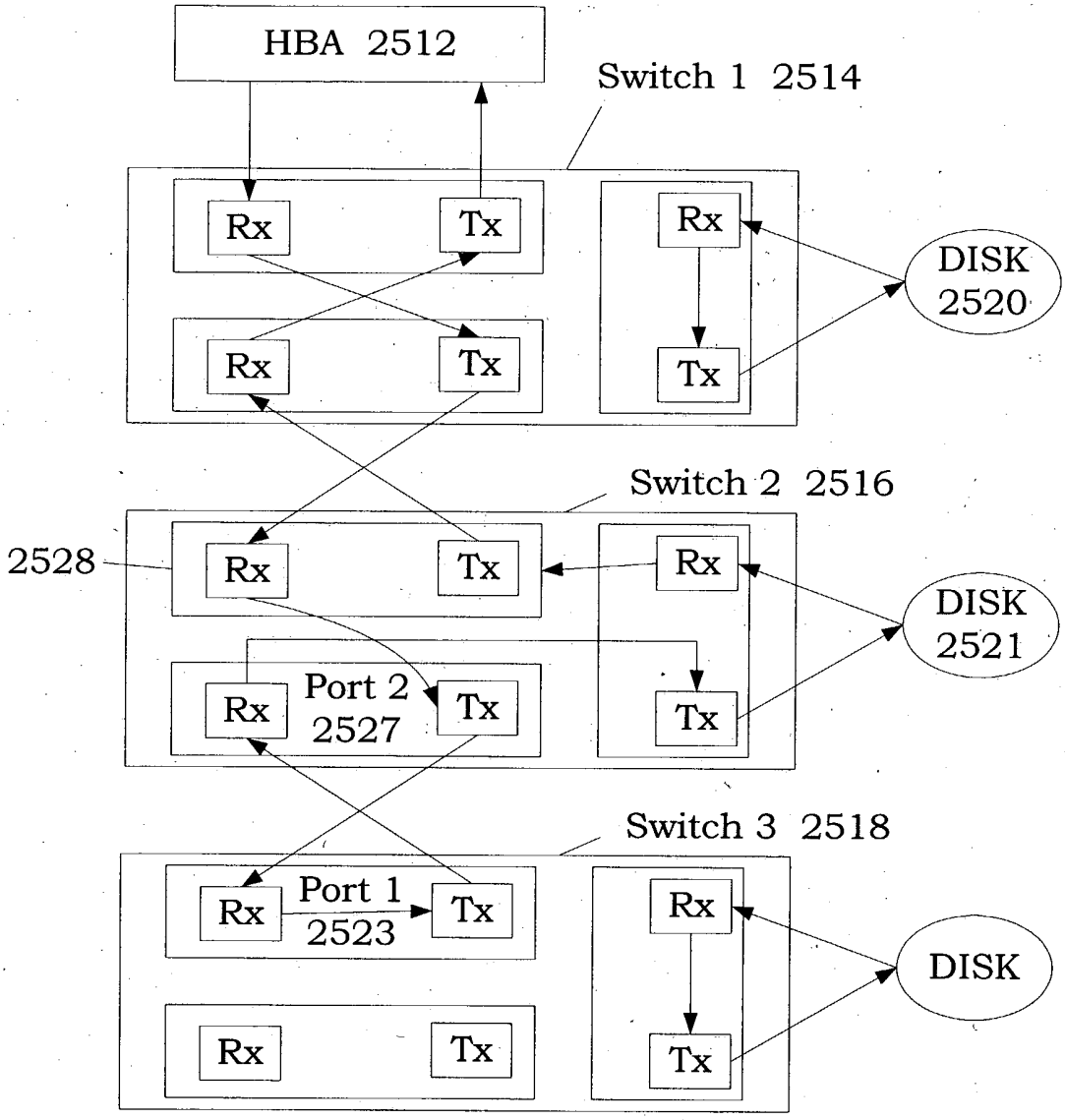

HBA 2512
Switch 1 2514
Rx
Tx
Rx
Tx
Rx
Tx
DISK
2520
Switch 2 2516
2528
Rx
Tx
Rx
DISK
2521
Rx
Port 2
2527
Tx
Tx
Switch 3 2518
Rx
Port 1
2523
Tx
Rx
DISK
Rx
Tx
Tx

Fig. 33

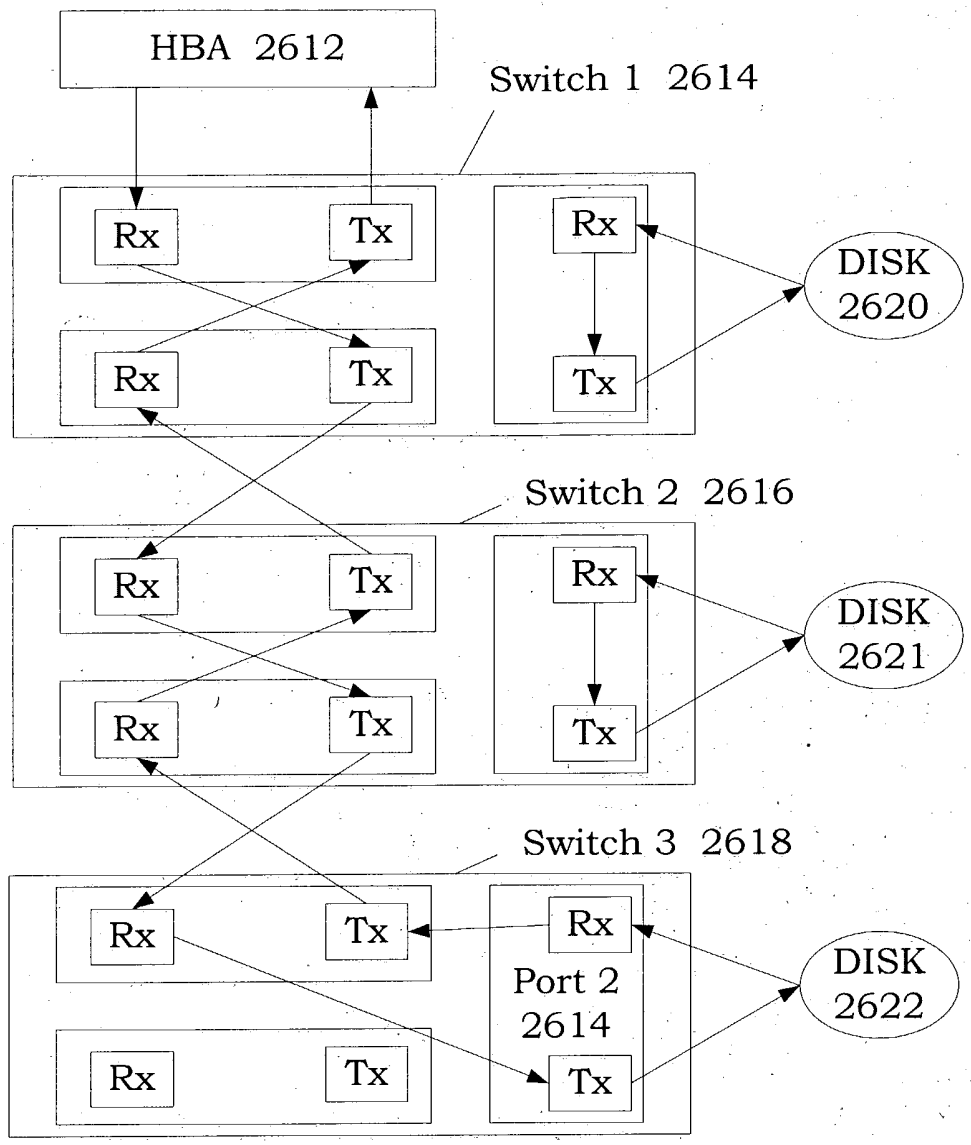

HBA 2612
Switch 1 2614
Rx
Tx
Rx
Tx
Rx
Tx
DISK 2620
Switch 2 2616
Rx
Tx
Rx
Tx
Rx
Tx
DISK 2621
Switch 3 2618
Rx
Tx
Rx
Tx
Rx
Tx
Port 2 2614
DISK 2622

Fig. 34

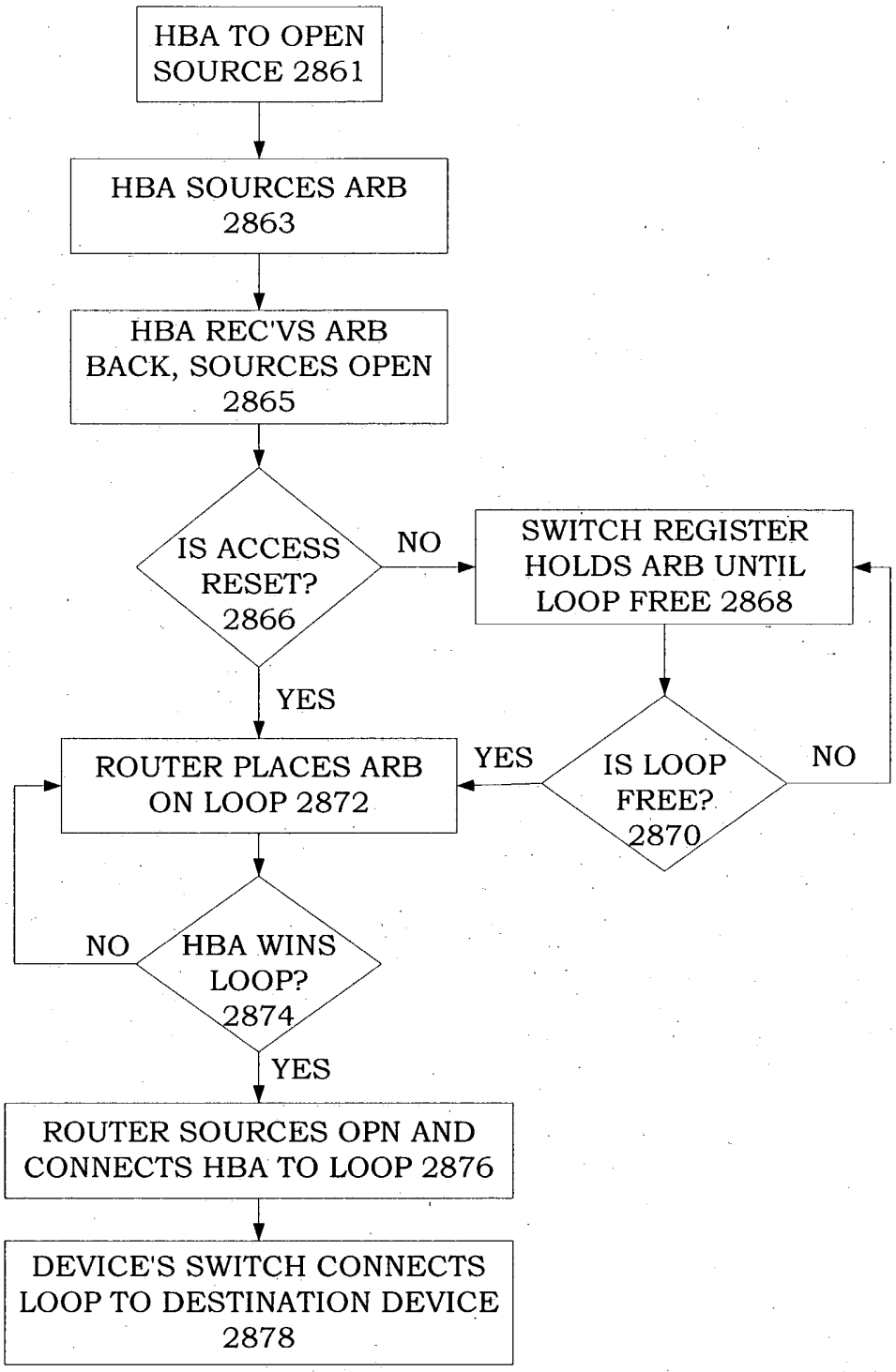

HBA TO OPEN SOURCE 2861
HBA SOURCES ARB 2863
HBA REC'VS ARB BACK, SOURCES OPEN 2865
IS ACCESS RESET? 2866
NO
SWITCH REGISTER HOLDS ARB UNTIL LOOP FREE 2868
YES
ROUTER PLACES ARB ON LOOP 2872
YES
IS LOOP FREE? 2870
NO
NO
HBA WINS LOOP? 2874
YES
ROUTER SOURCES OPN AND CONNECTS HBA TO LOOP 2876
DEVICE'S SWITCH CONNECTS LOOP TO DESTINATION DEVICE 2878

Fig. 35